The Complexity of Star-Freeness and the Free Conway Theories

Ph.D. Thesis

by

László Bernátsky

Supervisor

Zoltán Ésik

József Attila University, Szeged, Hungary

1999

Contents

Acknowledgements

I would like to thank Zoltán Ésik for encouragement and guidance, and Stephen L. Bloom for his kind hospitality during my stay at the Stevens Institute of Technology in Hoboken, New Jersey.

Introduction

This thesis is concerned with two problems: the problem of deciding if a regular language is star-free, and the description of the free Conway theories. Although these two topics seem to be quite far from each other, they both are strongly related to the automata-theoretic concept of *aperiodicity*.

The first chapter of the thesis is an introductory one. It does not contain any new results, only summarizes the notations and some well known results used in the forthcoming chapters.

The second chapter contains the new results on the complexity of star-freeness which were first published in [6]. The only exception is Theorem 2.3.2, which I only proved later and have not published yet. I also have to note here that, due to a minor modification of Construction 2.2.2, Theorem 2.3.1 is a slightly stronger than the corresponding theorem published in [6].

Star-free languages form an important subclass of regular languages: they are the ones that can be obtained from the singleton languages and the empty language by a finite number of applications of the operations of union, complement and product. By Schützenberger's [47] famous theorem, a regular language $L \subseteq \Sigma^*$ is star-free if and only if its syntactic monoid is aperiodic, i.e., if there exists an integer $k \geq 0$ such that for all words $u, v, w \in \Sigma^*$

$$uv^k w \in L \iff uv^{k+1} w \in L.$$

This is the same as to say that L is recognized by an aperiodic DFA. There also exists a logical characterization stating that a language is star-free if and only if it can be defined by a first-order formula of a suitable formal language, see [50]. Thus, it is an interesting question how hard is it to decide if a given regular language is star-free.

Naturally, when we are talking about deciding whether a language has some property we implicitly assume that the language is given by some finite representation. In case of a regular language this finite representation can be for example a deterministic or nondeterministic finite automaton, or a regular expression.

Depending on what representation we choose, we get different formalizations of the problem of deciding if a regular language is star-free. In this thesis we study the following decision problems (and some restricted versions theirof):

- Automata star-freeness (**ASF**): decide if a given nondeterministic finite au-

tomaton recognizes a star-free language.

- Regular expression star-freeness (**RSF**): decide if a given regular expression denotes a star-free language.

The first result on the complexity of **ASF** is due to Jacques Stern [49], who proved that the problem of deciding whether a *deterministic* finite automaton recognizes a star-free language is **coNP**-hard and belongs to **PSPACE**. A few years later Sang Cho and Dung T. Huynh [20] characterized the complexity of deciding whether a minimal deterministic automata having a two-element input alphabet recognizes a star-free language. They proved that already this severely restricted version of **ASF**, which we refer to as $\mathbf{ASF}_R$, is **PSPACE**-complete. Theorem 2.3.3 extends their result to nondeterministic automata by showing that the problem **ASF** is solvable in polynomial space, and therefore it is also **PSPACE**-complete. This result was published in [6]. The proof combinies Stern's original algorithm with the well known complexity theoretical idea that sometimes it is possible to decide some property of an exponentially large object (in our case, the minimal deterministic automaton which is language-equivalent to the given nondeterministic one) whithout actually constructing the object itself.

The same paper considered first the complexity of **RSF** and $\mathbf{RSF}_R$, where $\mathbf{RSF}_R$ is the restriction of the problem **RSF** to regular expressions of star-height two over a two element alphabet. It was proved that $\mathbf{RSF}_R$ is **PSPACE**-hard and **RSF** is solvable in polynomial space, so that both problems are **PSPACE**-complete. The proof of the **PSPACE**-hardness of $\mathbf{RSF}_R$ is very similar to the proof of the **PSPACE**-hardness of $\mathbf{ASF}_R$ given by Cho and Huynh, but it is based on a different construction of finite automata, see Construction 2.2.1. The proof given by Cho and Huynh strongly depends on Dexter Kozen's proof [39] for the **PSPACE**-hardness of the intersection problem of deterministic finite automata. The use of Construction 2.2.1 is not only essential in extending the proof of Cho and Huynh to regular expressions, but it also makes possible to strengthen Kozen's aforementioned result: in Theorem 2.3.1 we show that a severely restricted version of the automata intersection problem is already **PSPACE**-hard.

The third chapter is again an introductory chapter summarizing the notations and known results on many-sorted algebra. The algebra of flowchart schemes is introduced there as an example, and used later in the fourth chapter for describing the free Conway theories.

The algebraic study of flowchart schemes and flowchart algorithms was initiated in [27] and further developed in [13, 48, 19], to mention only a few references. Schemes may be defined as locally ordered, vertex labeled, finite digraphs with distinguished begin and exit nodes, each numbered by a nonnegative integer, so that each scheme has source n and target p for some nonnegative integers n, p. (We use $\mathbf{N}$ to denote the set of nonnegative integers.) The other nodes are consistently labeled by letters in a ranked alphabet, or signature. Schemes over a signature Σ are equipped with several constants and the operations of sequential composition, pairing or separated sum, which may be viewed as some sort of parallel composition, and a looping operation

called iteration. (The paper [19] uses feedback instead of iteration.) In [13], schemes over a signature Σ have been characterized as the free algebra generated by Σ in a variety of $\mathbf{N} \times \mathbf{N}$-sorted algebras axiomatized by a finite number of equation schemes or *meta-equations*, see page 43. See also [48, 19] for refinements of this result.

Σ-schemes with source n and target p may also be viewed as morphisms $n \to p$ in a small category whose objects are the nonnegative integers. Unless Σ is trivial, coproducts do not exist in this category, so that Σ-schemes do not form an algebraic theory in the sense of Lawvere [41]. Nevertheless, schemes are commonly interpreted in such theories which are enriched by a fixed-point operation modeling iteration. For example, the theories $\mathbf{Seq}_A$ of sequacious functions [25] on a set A are used to model the stepwise behavior of flowchart algorithms, while the theories of partial functions $\mathbf{Pfn}_A$ serve as semantic models for input-output behavior. Another common class of interpretations of schemes is as continuous functions over cpo's. In this approach, a scheme is a graphical representation of a recursive system of equations. When A is a cpo with a bottom element, each letter in Σ may be interpreted as a continuous function on A. Then the semantics of a Σ-scheme is a morphism in the theory $\mathbf{Th}_A$ of continuous functions over A, obtained as the least solution of the system of equations corresponding to the scheme.

The theories $\mathbf{Seq}_A$, $\mathbf{Pfn}_A$, and $\mathbf{Th}_A$ are all examples of *iteration theories*, which were originally defined in [11, 12] and [29]. The book [15] and the paper [16] give a summary of the results on iteration theories (and the properties of the fixed point operation in general).

It is known for example that the variety of iteration theories is generated by the theories $\mathbf{Seq}_A$, where A is a set, or by the theories $\mathbf{Th}_A$, where A is a cpo with a bottom element. (The theories of the form $\mathbf{Pfn}_A$ generate the subvariety consisting of the iteration theories with a unique morphism $1 \to 0$.) Thus, two schemes are equivalent under all interpretations in iteration theories (or *strongly equivalent*, for short) if and only if they are equivalent under all interpretations in the theories $\mathbf{Seq}_A$, or in the theories $\mathbf{Th}_A$. For this reason, iteration theories may be considered as the "standard" interpretations of flowchart schemes.

It is also well known that the equational theory of iteration theories (that is, the problem of deciding whether an equation holds in all iteration theories) is solvable in polynomial time. It is also decidable in polynomial time whether two schemes are strongly equivalent.

The fourth chapter of this thesis contains similar results about "nonstandard" interpretations of flowchart schemes, which were first published in [8]. By a nonstandard interpretation we mean a theory enriched with an iteration operation satisfying all equations true of flowcharts. One of the main results, Theorem 3.3.2 states that these theories are exactly the *Conway theories*, axiomatized by a small set of meta-equations including the well-known composition identity (3.48) which implies Elgot's fixed point equation (3.49). Thus the least congruence on Σ-schemes whose quotient is a theory gives the free Conway theory. The second main result, Theorem 4.3.2, provides a more explicit description of the free Conway theories. The description uses aperiodic simulations of flowchart schemes, a concept borrowed from automata

theory. See [44]. It follows that the equations that hold in Conway theories are exactly the valid "group-free" equations of iteration theories. Finally, we use the explicit description to prove that the Conway-equivalence problem of flowchart schemes is **PSPACE**-complete, cf. Theorem 4.3.3. It then follows that the equational theory of Conway theories is also **PSPACE**-complete. Theorems 3.3.2 and 4.3.2 answer open problems raised in [13] and [15].

Aside from serving as interpretation domains for flowchart schemes, our interest in (the free) Conway theories stems from several mathematical facts. First, the complete description of a variety of algebras should include (at least) an equational axiomatization, and also the description of the free algebras. For example, the papers [7, 31] give an axiomatization and a description of the free algebras for the variety generated by all algebras of binary relations with operations of union, composition, conversion and reflexive-transitive closure, and neutral elements 0 (the empty relation) and 1 (the identity relation). Second, the equational theory of iteration theories is axiomatized by the Conway theory axioms together with a complicated equation scheme, the commutative identity [29], or the group identities [30], or the Scott induction principle formulated to involve only equations [32]. (The second of the latter results may be seen as a generalization of Krob's result [40] on the axiomatization of the regular identities.) Comparing the structure of the free Conway theory with that of the free iteration theory, we obtain a clear picture of that part of the equational theory of iteration theories which is captured by the commutative identity, or the group identities. Also, our work explains the role of the commutative identity: it separates nonstandard models from the standard ones by equations. And finally, Conway theories are interesting in themselves for the following reasons.

- In a matrix theory [26, 15] equipped with a unary operation $a \mapsto a^*$, the Conway axioms are the two well-known sum and product identities

$$(a+b)^* = (a^*b)^*a^*$$
$$(ab)^* = a(ba)^*b + 1.$$

  Conway's book [21] contains many interesting identities which are consequences of just the Conway axioms. See also [40].

- A general Kleene-type theorem is a logical consequence of just the Conway axioms, see [15].

- It was proved in [14] that the soundness, and relative completeness of the Floyd-Hoare calculus in expressive models, is a consequence of the Conway theory axioms. Thus, even under nonstandard interpretations, one can reason about the correctness of flowchart programs using the Floyd-Hoare rules.

Chapter 1

Preliminaries

1.1 Sets and relations

The set of positive integers is denoted $[\omega]$ and the set of nonnegative integers is denoted $\mathbf{N}$. For each integer $n \in \mathbf{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$, so that $[0]$ is just another name for the empty set $\emptyset$. A set containing exactly one element is called a *singleton*.

Suppose that A and B are sets. Recall that the *power set* of A is the set $\mathrm{P}(A)$ consisting of all subsets of A, and the *direct product* of A and B is the set $A \times B$ consisting of all ordered pairs (a, b) with $a \in A$ and $b \in B$. A *(binary) relation from A to B* is just a subset of $A \times B$.[1] A relation from A to A is usually called a *relation on A*. For example, the set id_A of all pairs (a, a) with $a \in A$ is a relation on A, called the *identity relation on A*.

Suppose that ρ is a relation from A to B. The *inverse* of ρ is the relation

$$\rho^{-1} = \{(b, a) \in B \times A \mid (a, b) \in \rho\}.$$

Note that ρ^{-1} is a relation from B to A.

Suppose that A' is a subset of A and B' is a subset of B. The *image* of A' under ρ is the set

$$A'\rho = \{b \in B \mid \exists a \in A' \ (a, b) \in \rho\}.$$

The image of B' under ρ^{-1} is called the *inverse image* of B' under ρ. We write $A'\rho B'$ if $B' \subseteq A'\rho$ and $A' \subseteq B'\rho^{-1}$. If A' and B' are singletons, say $A' = \{a\}$ and $B' = \{b\}$, then we write $a\rho$ instead of $A'\rho$ and $a\rho b$ instead of $A'\rho B'$. Thus, $a\rho b$ is the same as to say that the pair (a, b) belongs to ρ.

The sets

$$\mathrm{Dom}(\rho) = \{a \in A \mid \exists b \in B \ \ a\rho b\}$$
$$\mathrm{Rng}(\rho) = \{b \in B \mid \exists a \in A \ \ a\rho b\}$$

[1] Thus, if A is a subset of A' and B is a subset B', then each relation from A to B is also a relation from A' to B'.

are called the *domain* and the *range* of ρ, respectively. The union of $\mathrm{Dom}(\rho)$ and $\mathrm{Rng}(\rho)$ is called the *carrier* of ρ, denoted $\mathrm{Car}(\rho)$. Note that ρ is a relation from $\mathrm{Dom}(\rho)$ to $\mathrm{Rng}(\rho)$, and it is also a relation on $\mathrm{Car}(\rho)$.

Suppose that τ is a relation from C to D. Then the *composite* of ρ and τ is the relation

$$\rho \circ \tau = \{(a, d) \in A \times D \mid \exists x \in B \cap C \; a\rho x \wedge x\tau d\}$$

from A to D. We usually write $\rho\tau$ instead of $\rho \circ \tau$. The *powers of the relation* ρ are defined by

$$\begin{aligned} \rho^0 &= \mathrm{id}_{\mathrm{Car}(\rho)}, \\ \rho^k &= \rho^{k-1}\rho, \\ \rho^{-k} &= (\rho^{-1})^k, \end{aligned}$$

for all integers $k > 0$. Note that $\rho^1 = \rho^0\rho = \rho\rho^0 = \rho$.

Suppose that θ is a relation on A. We say that θ is

reflexive if $\theta^0 \subseteq \theta$,

transitive if $\theta^2 \subseteq \theta$,

symmetric if $\theta^{-1} \subseteq \theta$,

antisymmetric if $\theta \cap \theta^{-1} = \emptyset$.

Moreover, θ is called a

preorder if it is reflexive and transitive,

partial order if it is reflexive, transitive and antisymmetric,

equivalence relation if it is reflexive, transitive and symmetric.

Suppose that θ is an equivalence relation on A, so that $A = \mathrm{Car}(\theta)$. By an *equivalence class* of θ we mean a nonempty subset S of A such that $S\theta = S$. The set of all equivalence classes of θ is called the *quotient* of A under θ, denoted A/θ. Note that

$$A/\theta = \{a\theta \mid a \in A\}$$

is a *partition* of the set A, i.e., it is a set of nonempty and pairwise disjoint subsets of A the union of which is A.

The relation $\rho \subseteq A \times B$ is called a *partial function* from A to B if for each element $a \in \mathrm{Dom}(\rho)$ there exists a unique element $b \in \mathrm{Rng}(\rho)$ (called the *image* of a under ρ) such that the pair (a, b) belongs to ρ. If in addition ρ is a *total relation* from A to B, i.e., $\mathrm{Dom}(\rho) = A$, then ρ is called a *function* from A to B. We write $\rho : A \to B$ to indicate that ρ is a (partial) function from A to B. A (partial) function from A

to A is also called a (partial) function on A. For example, the identity relation id_A is a function on A, therefore it is sometimes called the *identity function* on A.

Suppose that φ is a partial function from A to B. Then the image of an element $a \in \mathrm{Dom}(\varphi)$ is denoted $\varphi(a)$, sometimes φ_a.[2] The *kernel* of the partial function φ is the equivalence relation

$$\ker_\varphi = \{(x, y) \in \mathrm{Dom}(\varphi) \times \mathrm{Dom}(\varphi) \mid \varphi(x) = \varphi(y)\}$$

on $\mathrm{Dom}(\varphi)$. The partial function $\varphi : A \to B$ is called

constant if $\mathrm{Rng}(\varphi)$ is either empty or a singleton,

injective if $\ker_\varphi$ is the identity relation on $\mathrm{Dom}(\varphi)$,

surjective if $\mathrm{Rng}(\varphi) = B$,

bijective if it is injective and surjective.

An injective (respectively surjective, bijective) function from A to B is also called an *injection* (respectively *surjection*, *bijection*) from A to B. A bijective function on A is usually called a *permutation* on A.

We denote by

$\mathrm{Fn}[A, B]$ or B^A the set of all functions from A to B,

$\mathrm{Part}[A, B]$ the set of all *nonempty* partial functions from A to B,

$\mathrm{Const}[A, B]$ the set of all constant functions from A to B,

$\mathrm{Biject}[A, B]$ the set of all bijections from A to B.

Suppose that I and U are two sets and $A : I \to \mathrm{P}(U)$ is a function mapping each element $i \in I$ to a subset A_i of U. We call such a function A an *I-indexed family*, and we usually write $\{A_i\}_{i \in I}$ instead of $A : I \to \mathrm{P}(U)$. The *union*, *intersection* and *product* of the sets A_i are defined respectively by

$$\bigcup_{i \in I} A_i := \{s \in S \mid \exists i \in I \; s \in A_i\}$$
$$\bigcap_{i \in I} A_i := \{s \in S \mid \forall i \in I \; s \in A_i\}$$
$$\prod_{i \in I} A_i := \{f \in \mathrm{Fn}[I, U] \mid \forall i \in I \; f_i \in A_i\}.$$

[2] Note that the notations $\varphi(a)$ and φ_a make sense only if φ is a partial function and a is an element of $\mathrm{Dom}(\varphi)$, and in this case $a\varphi = \{\varphi(a)\} = \{\varphi_a\}$.

1.2 Words and languages

Suppose that A is a set. By a *finite word* of length $n \geq 0$ over A we mean a function from $[n]$ to A. The set of all finite words over A is denoted A^*, i.e.,

$$A^* = \bigcup_{n \in \mathbf{N}} A^{[n]}.$$

Note that the only element of $A^{[0]}$ is the empty function $\emptyset : \emptyset \to A$, which we call the *empty word* and denote by ϵ. The set of all nonempty finite words over A is denoted A^+, i.e., $A^+ = A^* \setminus \{\epsilon\}$.

By an *infinite word* over A we mean a function from $[\omega]$ to A, so that $A^{[\omega]}$ is the set of all infinite words over A.

We usually write A^n instead of $A^{[n]}$, and A^ω instead of $A^{[\omega]}$. When u is a (finite or infinite) word and $i \in \mathrm{Dom}(u)$, the image $u(i)$ of i under u is called the *ith letter of* u, which we prefer to denote by u_i. It is also convenient to identify each element $a \in A$ with the word of length 1 mapping 1 to a.

Suppose that $u, v \in A^* \cup A^\omega$ are words over A. The cardinality of $\mathrm{Dom}(u)$ is called the *length* of u and denoted by $|u|$. Thus, $\mathrm{Dom}(u) = [|u|]$. The *concatenation* of u and v is the word $u \cdot v$ defined by

$$i\,(u \cdot v) = \begin{cases} i\,u & \text{if } i \in \mathrm{Dom}(u) \\ (i - |u|)\,v & \text{otherwise,} \end{cases}$$

for all $i \in [\omega]$. Note that if u is an infinite word then $u \cdot v = u$. We usually omit the concatenation sign $\cdot$ and write uv instead of $u \cdot v$. This convention may cause some misunderstanding since the composite relation $u \circ v$ is also abbreviated uv. However, if u and v are words then $u \circ v$ usually doesn't make sense, and the reader should always think that uv stands for the word $u \cdot v$. Note that, according to our conventions, any finite word $w \in A^*$ can be written in the form $w_1 w_2 \cdots w_{|w|}$. A word $v \in A^*$ is called a *prefix* of a word $u \in A^* \cup A^\omega$ if $u = vw$, for some $w \in A^* \cup A^\omega$. A finite word $w \in A^*$ is called a *postfix* of a finite word $u \in A^*$ if u is of the form vw, for some word $v \in A^*$.

The powers of the word $u \in A^* \cup A^\omega$ are defined in the usual way:

$$u^0 = \epsilon$$
$$u^k = u^{k-1}u,$$

for all $k \in [\omega]$.

A *language* over A is just a subset of A^*. We extend the concatenation operation to languages by

$$L_1 \cdot L_2 = \{uv \mid u \in L_1,\ v \in L_2\},$$

for all languages $L_1, L_2 \subseteq A^*$. Just as for words, we usually omit the concatenation sign and write $L_1 L_2$ instead of $L_1 \cdot L_2$. The powers of a language $L \subseteq A^*$ are defined

by

$$L^0 = \{\epsilon\}$$
$$L^k = L^{k-1}L,$$

for all $k \in [\omega]$.

1.3 Many-sorted sets

Suppose that S is a set. By an *S-sorted set* we mean an S-indexed family $A = \{A_s\}_{s\in S}$ of sets. For any word $u \in S^n$ $(n \geq 0)$ we denote the set $A_{u_1} \cdots A_{u_n}$ by A_u. The *universe* of A is the set

$$\mathrm{Set}(A) = \bigcup_{s\in S} A_s,$$

and the *sort relation* $\mathrm{sort}_A \subseteq \mathrm{Set}(A) \times S$ associated with A is defined by

$$a \,\mathrm{sort}_A\, s \iff a \in A_s,$$

for all $a \in \mathrm{Set}(A)$ and $s \in S$. We say that an element $a \in \mathrm{Set}(A)$ *has sort* $s \in S$, or a *is of sort* S, if $a \in A_s$. Note that an element may have many sorts. The S-sorted set A is called

finite if $\mathrm{Set}(A)$ is finite,

infinite if $\mathrm{Set}(A)$ is infinite,

finitary if each of the sets A_s $(s \in S)$ is finite.

Note that a finitary many-sorted set may be infinite or finite. If sort_A is a function, we call it the *labeling function* of A, and denote it by label_A. In this case we call A an *S-labeled set*. In other words, an S-sorted set A is S-labeled if and only if $A_s \cap A_r = \emptyset$ holds for all *different* sorts $s, r \in S$. We shall think of S-labeled sets as simple (unsorted) sets with an associated labeling function. This means that if (and only if) A is an S-labeled set we shall simply write A instead of $\mathrm{Set}(A)$.

Suppose that A and B are S-sorted sets. We write $A \subseteq B$ and say that A is an *S-sorted subset of B* if $A_s \subseteq B_s$ holds for each $s \in S$. The usual set-theoretical operations of union, intersection and direct product are generalized to many-sorted sets in a straightforward way. For example, the *direct product* of A and B is the S-sorted set $A \times B$ defined by

$$(A \times B)_s = A_s \times B_s.$$

An *S-sorted relation from A to B* is an S-indexed family $\{\rho_s\}_{s\in S}$ of relations such that $\rho_s \subseteq A_s \times B_s$, for each $s \in S$. Note that an S-sorted relation from A to B is just

an S-sorted subset of $A \times B$. We call the S-sorted relation ρ reflexive (respectively transitive, (anti)symmetric) if each of the relations ρ_s has the corresponding property. The concepts of *many-sorted preorder*, *partial order* and *equivalence* are defined in the same way as for ordinary relations.

Suppose that $\theta \subseteq A \times A$ is an S-sorted equivalence relation on A. Then the *quotient* of A under θ is the S-sorted set A/θ with

$$(A/\theta)_s = A_s/\theta_s,$$

for all $s \in S$.

Suppose that I is a set of indices, and A_i is an S-sorted set, for each $i \in I$. Then the *product* of the A_i is the S-sorted set

$$P = \prod_{i \in I} A_i$$

such that for all $s \in S$

$$P_s = \prod_{i \in I} (A_i)_s.$$

Thus, the elements of P_s are all functions $f : I \to \bigcup_{i \in I} \mathrm{Set}(A_i)$ such that $f(i) \in (A_i)_s$, for all $i \in I$ and $s \in S$.

Suppose now that A is a simple (not sorted) set. By an *operation on* A we mean a function $A^n \to A$, for some $n \in \mathbf{N}$. The operations on A form an $\mathbf{N}$-labeled set Opn(A) defined by

$$\mathrm{Opn(A)}_n = \mathrm{Fn}[A^n, A],$$

for all $n \in \mathbf{N}$. We extend this notation to the case when A is an S-sorted set: in this case Opn(A) is the $S^* \times S$-sorted set with

$$\mathrm{Opn(A)}_{(u,s)} = \mathrm{Fn}[A_u, A_s],$$

for all $u \in S^*$ and $s \in S$.

1.4 Finite automata

Most of our automata-theoretical notations and definitions are adopted from [24].

A *nondeterministic finite automaton* (*NFA*) is a 5-tuple $\mathcal{A} = (Q, \Sigma, \tau, I, F)$, where

- Q is the finite set of states,
- Σ is the finite set of input symbols,
- $\tau : \Sigma \to \mathrm{P}(Q \times Q)$ is the transition function,

- $I \subseteq Q$ is the set of initial states,
- $F \subseteq Q$ is the set of final states.

Thus for each input symbol $\sigma \in \Sigma$, $\tau(\sigma)$ is a binary relation on Q, called the *relation induced by σ in the automaton $\mathcal{A}$*. We prefer the notation $\sigma_{\mathcal{A}}$ to $\tau(\sigma)$. When $u \in \Sigma^*$ is an input word, $u_{\mathcal{A}}$ denotes the *relation induced by u in $\mathcal{A}$*, defined by

$$u_{\mathcal{A}} := \tau(u_1) \circ \cdots \circ \tau(u_{|u|}).$$

Note that $\epsilon_{\mathcal{A}}$ is the identity relation on Q.

The automaton $\mathcal{A}$ can be visualized as a directed graph with vertices Q, and edges labeled by input symbols in Σ. Motivated by this point of view, we shall sometimes write $q \xrightarrow{u}_{\mathcal{A}} q'$ to indicate that there is a directed u-labeled walk from vertex q to vertex q', which is the same as to write $q\, u_{\mathcal{A}}\, q'$. For sets S, S' of states the notation $S \xrightarrow{u}_{\mathcal{A}} S'$ means that there exist some states $q \in S$ and $q' \in S'$ such that $q \xrightarrow{u}_{\mathcal{A}} q'$.[3]

The *language $L(\mathcal{A})$ recognized by $\mathcal{A}$* consists of those words $u \in \Sigma^*$ for which there exists a u-labeled walk from some initial state to a final state, formally

$$L(\mathcal{A}) = \{u \in \Sigma^* \mid I \xrightarrow{u}_{\mathcal{A}} F\}.$$

When $\mathcal{A}$ is understood we omit the subscript in $\xrightarrow{u}_{\mathcal{A}}$ and $u_{\mathcal{A}}$.

We call $\mathcal{A}$ a *deterministic finite automaton* (*DFA*) if it has at most one initial state, and each relation $\sigma_{\mathcal{A}}$ ($\sigma \in \Sigma$) is a partial function $Q \to Q$. A deterministic automaton is called *complete* if it has a unique initial state, and each of its input symbols induces a total function on Q. The automaton $\mathcal{A}$ is called a *reset automaton* if it has a unique initial state and each input symbol $\sigma \in \Sigma$ induces either the identity function or a *partial constant function on Q*, that is, a partial function $f : Q \to Q$ with $|\mathrm{Rng}(f)| \leq 1$. A *1-reset automaton* is a reset automaton with a unique final state in which the *inverse* of each relation induced by an input symbol is either the identity function or a partial constant function. In other words, a 1-reset automaton has a unique initial state and a unique final state, and each input symbol induces either the identity function or a singleton relation or the empty relation on its states.

A state q of $\mathcal{A}$ is called *accessible* (respectively, *coaccessible*) if there exists some input word $u \in \Sigma^*$ with $I \xrightarrow{u}_{\mathcal{A}} \{q\}$ (respectively, $\{q\} \xrightarrow{u}_{\mathcal{A}} F$). Note that each initial state is accessible and each final state is coaccessible. A *biaccessible* state is one which is both accessible and coaccessible. Two states $q, q' \in Q$ are called *equivalent*, denoted $q \approx_{\mathcal{A}} q'$, if for all input words $u \in \Sigma^*$

$$\{q\} \xrightarrow{u}_{\mathcal{A}} F \iff \{q'\} \xrightarrow{u}_{\mathcal{A}} F.$$

Suppose that $\mathcal{A}$ is a DFA. Then $\mathcal{A}$ is called

[3]Note that the two notations $S \xrightarrow{u}_{\mathcal{A}} S'$ and $S\, u_{\mathcal{A}} S'$ have different meaning, because the latter means that each element of S is related to some element of S' by $u_{\mathcal{A}}$, and each element of S' is related to some element of S by ${u_{\mathcal{A}}}^{-1}$, see Section 1.1.

minimal if all of its states are biaccessible, and it has no different equivalent states,

aperiodic if there exists an integer $k \geq 0$ such that $(u^k)_{\mathcal{A}} = (u^{k+1})_{\mathcal{A}}$, for all $u \in \Sigma^*$.

Observe that if $\mathcal{A}$ is a reset automaton then $(u^2)_{\mathcal{A}} = (u^3)_{\mathcal{A}}$, and if $\mathcal{A}$ is a complete reset automaton then $u_{\mathcal{A}} = (u^2)_{\mathcal{A}}$, for all words $u \in \Sigma^*$.

Remark 1.4.1. It is well known (see [24]) that a deterministic automaton $\mathcal{A} = (Q, \Sigma, \tau, I, F)$ is aperiodic if and only if it satisfies the implication

$$q \xrightarrow{u^k}_{\mathcal{A}} q \Longrightarrow q \xrightarrow{u}_{\mathcal{A}} q,$$

for all states $q \in Q$, input words $u \in \Sigma^+$ and integers $k \geq 2$.

Suppose that $n \geq 1$, and $\mathcal{A}_i = (Q_i, \Sigma, \tau_i, I_i, F_i)$ is an NFA, for each $i \in [n]$. Then the *product* of the A_i is the NFA

$$\prod_{i\in[n]} \mathcal{A}_i = (\prod_{i\in[n]} Q_i,\ \Sigma,\ \tau,\ \prod_{i\in[n]} I_i,\ \prod_{i\in[n]} F_i),$$

where

$$\tau(\sigma) = \{((q_1, \ldots, q_n), (r_1, \ldots, r_n)) \mid \forall i \in [n]\ (q_i, r_i) \in \tau_i(\sigma)\},$$

for all $\sigma \in \Sigma$. It is easy to see that

$$L(\prod_{i\in[n]} \mathcal{A}_i) = \bigcap_{i\in[n]} L(\mathcal{A}_i).$$

1.5 Turing machines

A *deterministic Turing machine* (*DTM*) with a single one-way infinite tape is a system $\mathcal{M} = (Q, \Gamma, \Sigma, \delta, q_0, q_f)$, where

- Q is the finite set of states,
- Γ is the finite set of tape symbols containing the special "blank" symbol $\flat$,
- $\Sigma \subseteq \Gamma$ is the finite set of input symbols, $\flat \notin \Sigma$,
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, 0, 1\}$ is the partial transition function,
- $q_0 \in Q$ is the initial state,
- $q_f \in Q$ is the final state.

We say that the machine $\mathcal{M}$ is in the *configuration* (q, i, u) for a state $q \in Q$, integer $i \in [\omega]$ and infinite word $u \in \Gamma^\omega$ if in state q it scans the ith tape cell and the

content of the tape is u. We define a binary relation $\vdash_{\mathcal{M}}$ on the set $Q \times [\omega] \times \Gamma^{\omega}$ of configurations by

$$(q, i, u) \vdash_{\mathcal{M}} (r, j, v) \iff \delta(q, u_i) = (r, v_i, j - i) \ \wedge \ \forall t \in [\omega] \ (t \neq i \implies v_t = u_t).$$

Note that $\vdash_{\mathcal{M}}$ is a partial function. The machine $\mathcal{M}$ *accepts* an input word $u \in \Sigma^*$ if

$$(q_0, 1, ub^{\omega}) \vdash_{\mathcal{M}}^{*} (q_f, 1, b^{\omega}),$$

otherwise $\mathcal{M}$ *rejects* u. The *language* $L(\mathcal{M}) \subseteq \Sigma^*$ *recognized by* $\mathcal{M}$ consists of those words $u \in \Sigma^*$ which are accepted by $\mathcal{M}$. Thus, for each word $u \in L(\mathcal{M})$, there exists a shortest sequence $(q_1, i_1, w_1), \ldots, (q_k, i_k, w_k)$ of configurations such that

$$\begin{aligned} (q_1, i_1, w_1) &= (q_0, 1, ub^{\omega}), \\ (q_k, i_k, w_k) &= (q_f, 1, b^{\omega}), \qquad \text{and} \\ (q_t, i_t, w_t) &\vdash_{\mathcal{M}} (q_{t+1}, i_{t+1}, w_{t+1}), \end{aligned}$$

for all $t \in [k-1]$. Then we define

$$\mathrm{SPACE}_{\mathcal{M}}(u) := \max_{t \in [k]} i_t.$$

Suppose that $S : \mathbf{N} \to \mathbf{N}$ is a function. The machine $\mathcal{M}$ is said to have *space complexity* S if $\mathrm{SPACE}_{\mathcal{M}}(u) \leq S(|u|)$, for all words $u \in L(\mathcal{M})$. The language class **PSPACE** consists of those languages which are recognized by some Turing machine $\mathcal{M}$ having space-complexity p, for some polynomial function $p : \mathbf{N} \to \mathbf{N}$.[4]

Suppose that L and L' are languages. We shall write $L \leq_{log} L'$ to indicate that L is logspace-reducible to L'.[5] The language L is called **PSPACE**-hard with respect to logspace-reductions, written **PSPACE** $\leq_{log} L$, if each language in **PSPACE** is logspace-reducible to L. Lastly, L is called **PSPACE**-complete with respect to logspace-reductions if $L \in$ **PSPACE** and **PSPACE** $\leq_{log} L$.

1.6 Universal algebra

For the reader's convenience we recall some basic definitions and facts of universal algebra.

Suppose that Σ is a *signature*, i.e., an $\mathbf{N}$-labeled set of *operation symbols*. The elements of Σ_n ($n \in \mathbf{N}$) are called n-ary operation symbols. A Σ-*algebra* is a pair $\mathcal{A} = (A, \tau)$, where A is a nonempty set (the *universe* of $\mathcal{A}$) and $\tau : \Sigma \to \mathrm{Opn(A)}$ is an $\mathbf{N}$-sorted function mapping each operation symbol $\sigma \in \Sigma_n$ ($n \in \mathbf{N}$) to a function $\sigma^{\mathcal{A}} : A^n \to A$.

Suppose that X is a set disjoint from Σ. The set of Σ-*terms with variables in* X is the least set $\mathrm{T}_{\Sigma}(X)$ satisfying

[4] This is not the standard definition, but is does not matter for **PSPACE**.

[5] We assume the reader is familiar with the concept of logspace-reducibility (see [17] or [43], for example).

1. $X \cup \Sigma_0 \subseteq \mathrm{T}_\Sigma(X)$.

2. If $t_1, \ldots, t_n \in \mathrm{T}_\Sigma(X)$ and $\sigma \in \Sigma_n$ for some $n \geq 1$ then $\sigma(t_1, \ldots, t_n) \in \mathrm{T}_\Sigma(X)$.

Observe that $\mathrm{T}_\Sigma(X) = \emptyset$ if and only if $X = \Sigma_0 = \emptyset$.

Suppose now that X is a set such that $\mathrm{T}_\Sigma(X) \neq \emptyset$. Then we denote by $\mathcal{T}_\Sigma(X)$ the *Σ-algebra of Σ-terms* with the operations defined in the natural way:

$$\sigma^{\mathcal{T}_\Sigma(X)}(t_1, \ldots, t_n) = \sigma(t_1, \ldots, t_n),$$

for all terms $t_1, \ldots, t_n \in \mathrm{T}_\Sigma(X)$.

Suppose that $\mathcal{A} = (A, \alpha)$ and $\mathcal{B} = (B, \beta)$ are Σ-algebras. A relation $\rho \subseteq A \times B$ is called a *simulation*[6] *from $\mathcal{A}$ to $\mathcal{B}$* if

$$a_1 \,\rho\, b_1 \;\wedge \cdots \wedge\; a_n \,\rho\, b_n \Longrightarrow \sigma^{\mathcal{A}}(a) \;\rho\; \sigma^{\mathcal{B}}(b),$$

for all integers $n \in \mathbf{N}$, symbols $\sigma \in \Sigma_n$ and words $a \in A^n$, $b \in B^n$. If in addition ρ is a function from A to B then we call it a *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$. A surjective homomorphism is called an *epimorphism*, and a bijective homomorphism is called an *isomorphism*. We say that $\mathcal{B}$ is a *homomorphic image* of $\mathcal{A}$ if there exists an epimorphism from $\mathcal{A}$ to $\mathcal{B}$. The two algebras $\mathcal{A}$ and $\mathcal{B}$ are called *isomorphic*, denoted $\mathcal{A} \simeq \mathcal{B}$, if there exists an isomorphism from $\mathcal{A}$ to $\mathcal{B}$. A *congruence* on $\mathcal{A}$ is an equivalence relation θ on A such that θ is a simulation from $\mathcal{A}$ to $\mathcal{A}$. We denote by $\mathrm{Con}(\mathcal{A})$ the set of all congruences on $\mathcal{A}$. We say that $\mathcal{B}$ is a *subalgebra* of $\mathcal{A}$ if $B \subseteq A$ and id_B is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. A (nonempty) subset of A is called a *subuniverse* of $\mathcal{A}$ if it is the universe of a subalgebra of $\mathcal{A}$. Note that each subalgebra of $\mathcal{A}$ is totally determined by its universe.

Suppose that A_0 is a subset of A such that at least one of the sets A_0 and Σ_0 is not empty. Then there exists a smallest subuniverse $\langle A_0 \rangle_{\mathcal{A}}$ of $\mathcal{A}$ containing A_0, called the *subuniverse generated by A_0 in $\mathcal{A}$*. The unique subalgebra of $\mathcal{A}$ with universe $\langle A_0 \rangle_{\mathcal{A}}$ is called the *subalgebra generated by A_0* and denoted also by $\langle A_0 \rangle_{\mathcal{A}}$. We say that A_0 is a *generating system* of $\mathcal{A}$, or equivalently, $\mathcal{A}$ is *generated by A_0*, if $\langle A_0 \rangle_{\mathcal{A}} = \mathcal{A}$.

Suppose that θ is a congruence on $\mathcal{A} = (A, \alpha)$. The *quotient of $\mathcal{A}$ under θ* is the Σ-algebra $\mathcal{A}/\theta$ with universe A/θ and operations defined by

$$\sigma^{\mathcal{A}/\theta}((a_1\theta) \cdots (a_n\theta)) = \sigma^{\mathcal{A}}(a)\theta$$

for all $n \geq 0$, $\sigma \in \Sigma_n$ and $a \in A^n$.

Suppose that I is a set of indices, and $\mathcal{A}_i = (A_i, \alpha_i)$ is a Σ-algebra, for each $i \in I$. The *product* of the $\mathcal{A}_i$ is the Σ-algebra

$$\mathcal{P} = \prod_{i \in I} \mathcal{A}_i$$

[6] The name "simulation" is borrowed from automata theory.

with universe $P = \prod_{i \in I} A_i$, and operations defined by

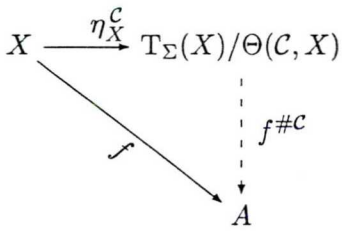
$$(\sigma^{\mathcal{P}}(f))(i) = \sigma^{\mathcal{A}_i}(f_1(i) \cdots f_n(i)),$$

for all $n \geq 0$, $\sigma \in \Sigma_n$, $i \in I$ and $f \in P^n$.

Suppose that $\mathcal{C}$ is a class of Σ-algebras. Then $\mathbf{H}(\mathcal{C})$, $\mathbf{S}(\mathcal{C})$ and $\mathbf{P}(\mathcal{C})$ respectively denote the classes of all homomorphic images, subalgebras and products of algebras in $\mathcal{C}$. The class $\mathcal{C}$ is called a *variety* (cf. [18]) of Σ-algebras if it is closed under the operations $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{P}$, i.e., if $\mathbf{H}(\mathcal{C}) \subseteq \mathcal{C}$, $\mathbf{S}(\mathcal{C}) \subseteq \mathcal{C}$ and $\mathbf{P}(\mathcal{C}) \subseteq \mathcal{C}$. The *variety generated by* $\mathcal{C}$ is the smallest variety $\mathbf{V}(\mathcal{C})$ containing $\mathcal{C}$. It is well known that

$$\mathbf{V}(\mathcal{C}) = \mathbf{HSP}(\mathcal{C}).$$

Suppose that X is a set, Σ is a signature with $\Sigma_0 \neq \emptyset$, and $\mathcal{C}$ is a class of Σ-algebras. We denote by $\mathcal{F}_{\mathcal{C}}(X)$ the Σ-algebra $\mathrm{T}_\Sigma(X)/\Theta(\mathcal{C}, X)$, where $\Theta(\mathcal{C}, X)$ is the intersection of the kernels of all homomorphisms $h : \mathrm{T}_\Sigma(X) \to \mathcal{A}$, for all algebras $\mathcal{A} \in \mathcal{C}$. Then $\mathcal{F}_{\mathcal{C}}(X)$ has the property that for each algebra $\mathcal{A} = (A, \alpha)$ in $\mathcal{C}$, and for each function $f : X \to A$, there exists a unique homomorphism $f^{\#\mathcal{C}} : \mathcal{F}_{\mathcal{C}}(X) \to \mathcal{A}$ such that the diagram



commutes (that is, $f^{\#\mathcal{C}}(\eta_X^{\mathcal{C}}(x)) = f(x)$, for all $x \in X$), where $\eta_X^{\mathcal{C}}$ is the function mapping each element $x \in X$ to the congruence class $x\,\Theta(\mathcal{C}, X)$. When $\mathcal{C}$ is a variety the algebra $\mathcal{F}_{\mathcal{C}}(X)$ itself belongs to $\mathcal{C}$. In this case we call $\mathcal{F}_{\mathcal{C}}(X)$ *the X-generated free algebra in $\mathcal{C}$*. Since we identify isomorphic algebras, any algebra isomorphic to $\mathcal{F}_{\mathcal{C}}(X)$ is also called the X-generated free algebra in $\mathcal{C}$. In particular, if $\mathcal{C}$ is the class of all Σ-algebras then the X-generated free algebra in $\mathcal{C}$ is isomorphic to the term algebra $\mathrm{T}_\Sigma(X)$. In this case we write $f^{\#}$ for $f^{\#\mathcal{C}}$.

Suppose that X is a set of variables. By an *equation with variables in X* we mean an ordered pair (t_1, t_2) of Σ-terms $t_1, t_2 \in \mathrm{T}_\Sigma(X)$, usually written in the form $t_1 = t_2$. We say that *the equation $t_1 = t_2$ is valid in the algebra $\mathcal{A} = (A, \alpha)$*, or that the algebra $\mathcal{A}$ *satisfies the equation* $t_1 = t_2$, or $\mathcal{A}$ is a *model of the equation* $t_1 = t_2$, denoted $\mathcal{A} \models t_1 = t_2$, if $f^{\#}(t_1) = f^{\#}(t_2)$ holds for all functions $f : X \to A$.

Suppose that $\mathcal{C}$ is a class of Σ-algebras and E is a set of equations. We write $\mathcal{C} \models E$ if each algebra in $\mathcal{C}$ satisfies each equation in E. We denote by $\mathrm{Eq}_X(\mathcal{C})$ the set of those equations with variables in X which are valid in each algebra in $\mathcal{C}$. The *equational theory* of the class $\mathcal{C}$ is the set $\mathrm{Eq}(\mathcal{C}) := \mathrm{Eq}_{\mathbf{X}}(\mathcal{C})$, where $\mathbf{X}$ is a fixed countably infinite set of variables, say $\mathbf{X} = \{x_1, x_2, \dots\}$. We let $\mathrm{Mod}(E)$ denote the class of those Σ-algebras which satisfy all equations in E. An equation e is called a *logical consequence* of E, denoted $E \models e$, if $\mathrm{Mod}(E) \models e$. Note that $\mathrm{Eq}(\mathrm{Mod}(E))$ is

the set of all logical consequences of E. We say that E *(equationally) axiomatizes* the class $\mathcal{C}$ if $\mathcal{C} = \mathrm{Mod}(E)$. Conversely, $\mathcal{C}$ is called an *equational class* of Σ-algebras if it is axiomatized by a set of equations. Note that $\mathcal{C}$ is equational if and only if $\mathcal{C} = \mathrm{Mod}(\mathrm{Eq}(\mathcal{C}))$. Birkhoff's famous theorem (originally published in [9]) states that a class of Σ-algebras is equational if and only if it is a variety.

Suppose that $t, t' \in \mathrm{T}_\Sigma(X)$ are Σ-terms and $x \in X$ is a variable. Then $t[t'/x]$ denotes the Σ-term which we get by substituting the term t' for each occurrence of x in t. The *deductive closure* of a set E of equations is the least set $\mathrm{D}(E)$ of equations such that for all $x \in X$

$$\begin{aligned} e \in E &\Longrightarrow e \in \mathrm{D}(E) \\ t \in \mathrm{T}_\Sigma(X) &\Longrightarrow t = t \in \mathrm{D}(E) \\ t_1 = t_2 \in \mathrm{D}(E) &\Longrightarrow t_2 = t_1 \in \mathrm{D}(E) \\ t_1 = t_2 \in \mathrm{D}(E) \ \wedge\ t_2 = t_3 \in \mathrm{D}(E) &\Longrightarrow t_1 = t_3 \in \mathrm{D}(E) \\ t_1 = t_2 \in \mathrm{D}(E) \ \wedge\ t_3 = t_4 \in \mathrm{D}(E) &\Longrightarrow t_1[t_3/x] = t_2[t_4/x] \in \mathrm{D}(E) \end{aligned}$$

An equation e is called a *syntactical consequence* of a set E of equations, denoted $E \vdash e$, if $e \in \mathrm{D}(E)$. It was also proved by Birkhoff in [9] that the notions of logical and syntactical consequence are the same, i.e., $E \models e \iff E \vdash e$.

1.6.1 Regular expressions

Let $\mathcal{R}$ be the signature consisting of the constant symbol $\emptyset$, the unary symbols $^*, \sim$ and the binary symbols $\cdot, \cup, \cap$. Suppose that A is a set disjoint from $\mathcal{R}$, and $\mathcal{R}'$ is a subset of $\mathcal{R}$. By an *$\mathcal{R}'$-type regular expression over A* we mean an $\mathcal{R}'$-term with variables in A. A $\{\emptyset, \cdot, \cup, ^*\}$-type regular expression is simply called a *regular expression*, and an $(\mathcal{R} \setminus \{^*\})$-type regular expression is also called a *star-free regular expression*.

As for the syntactic conventions, we use infix notation for the binary operations $\cup, \cap$ and $\cdot$, postfix notation for *, and we write $\overline{a}$ instead of $\sim a$. The operation symbol $\cdot$ is usually omitted. If $A' = \{a_1, \ldots, a_n\}$ is a subset of A, we simply write A' instead of $a_1 \cup \cdots \cup a_n$.

The language $L(E) \subseteq A^*$ denoted by an $\mathcal{R}$-type regular expression E over A is defined in the straightforward way, see [45].

A language $L \subseteq A^*$ is called *star-free* if it is denoted by some star-free regular expression over A.

We recall from [45] that the *star-height* $\mathrm{sh}(E)$ of a regular expression E over A is

defined by

$$\begin{aligned} \mathrm{sh}(a) &= 0 \\ \mathrm{sh}(\emptyset) &= 0 \\ \mathrm{sh}(E \cup F) &= \max\{\mathrm{sh}(E), \mathrm{sh}(F)\} \\ \mathrm{sh}(E \cdot F) &= \max\{\mathrm{sh}(E), \mathrm{sh}(F)\} \\ \mathrm{sh}(E^*) &= 1 + \mathrm{sh}(E), \end{aligned}$$

for all letters $a \in A$ and regular expressions E, F over A.

Chapter 2

The complexity of star-freeness

This chapter contains the new results on the complexity of star-freeness which were first published in [6]. There is one result, Theorem 2.3.2, which I proved later and have not published yet. I also have to note here that, due to a minor modification of Construction 2.2.2, Theorem 2.3.1 is a slightly stronger than the corresponding theorem published in [6].

2.1 Problems

We are going to characterize the computational complexity of the following decision problems.

1. The automata intersection problem (**AIP**):

   INSTANCE: A sequence $\mathcal{A}_1, \ldots, \mathcal{A}_n$ $(n \geq 2)$ of nondeterministic finite automata with a common set of input symbols.

   QUESTION: Does $\bigcap_{i \in [n]} L(\mathcal{A}_i) \neq \emptyset$ hold?

2. The intersection problem of minimal 1-reset automata ($\mathbf{AIP}_R$):

   INSTANCE: A sequence $\mathcal{A}_1, \ldots, \mathcal{A}_n$ $(n \geq 2)$ of minimal 1-reset automata with a common set of input symbols.

   QUESTION: Does $\bigcap_{i \in [n]} L(\mathcal{A}_i) \neq \emptyset$ hold?

3. The intersection problem of complete reset automata ($\mathbf{AIP}_C$):

   INSTANCE: A sequence $\mathcal{A}_1, \ldots, \mathcal{A}_n$ $(n \geq 2)$ of complete reset automata with a common set of input symbols.

   QUESTION: Does $\bigcap_{i \in [n]} L(\mathcal{A}_i) \neq \emptyset$ hold?

4. Automaton star-freeness (**ASF**):

   INSTANCE: A nondeterministic finite automaton $\mathcal{A}$.

QUESTION: Does $\mathcal{A}$ recognize a star-free language?

5. A restricted version of automaton star-freeness ($\mathbf{ASF}_R$):

   INSTANCE: A minimal DFA $\mathcal{A}$ with input symbols {0,1}.

   QUESTION: Does $\mathcal{A}$ recognize a star-free language?

6. Regular expression star-freeness (**RSF**):

   INSTANCE: A regular expression E.

   QUESTION: Does E denote a star-free language?

7. A restricted version of regular expression star-freeness ($\mathbf{RSF}_R$):

   INSTANCE: A regular expression E of star-height 2 over the 2-element set $\{0,1\}$.

   QUESTION: Does E denote a star-free language?

Assuming some efficient encoding of automata and regular expressions (see [43, 33]) with words over a fixed finite set of symbols, all these problems can be considered as languages. We are going to prove

Proposition 2.1.1. *The problems* **AIP**, $\mathbf{AIP}_R$, **ASF**, $\mathbf{ASF}_R$, **RSF** *and* $\mathbf{RSF}_R$ *are* **PSPACE***-complete with respect to logspace reductions. The problem* $\mathbf{AIP}_C$ *is solvable in polynomial time.*

2.2 Constructions

In this section we present the constructions of automata and regular expressions which are needed to show that the restricted problems $\mathbf{AIP}_R$, $\mathbf{ASF}_R$ and $\mathbf{RSF}_R$ are **PSPACE**-hard.

Each construction is divided into four parts titled *Input*, *Output*, *Description*, and *Proof*. The *Input* and *Output* parts respectively describe the preconditions and postconditions of the construction, which itself is described in the third part. In the fourth part we prove that if we apply the construction to some input data satisfying the preconditions then the resulting data will satisfy the postconditions. We should also prove that, assuming some efficient encoding of the input and output as words over some finite alphabet, the construction can be carried out by a logspace-bounded Turing machine. However, this is obviously true for all of our constructions, so we omit the proofs.

Our first construction shows how can one "replace" a deterministic Turing machine with a sequence of 1-reset automata.

Construction 2.2.1. *Input.* A polynomial function $p : \mathbf{N} \to \mathbf{N}$, a DTM $\mathcal{M} = (Q, \Gamma, \Sigma, \delta, q_0, q_f)$ of space-complexity p, and an input word $u \in \Sigma^n$, $n \geq 0$.

Output. A sequence $\mathcal{S}, \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_m$ of 1-reset automata, where $m = p(n)$, and

$$u \in L(\mathcal{M}) \iff L(\mathcal{S}) \cap L(\mathcal{P}) \cap \bigcap_{i \in [m]} L(A_i) \neq \emptyset. \tag{2.1}$$

Description. Let

$$\begin{aligned} \mathcal{S} &= (Q, A, \tau_{\mathcal{S}}, \{q_0\}, \{q_f\}) \\ \mathcal{P} &= ([m], A, \tau_{\mathcal{P}}, \{1\}, \{1\}), \end{aligned}$$

and for each $i \in [m]$,

$$\mathcal{A}_i = (\Gamma, A, \tau_i, \{(u\flat^\omega)_i\}, \{\flat\}),$$

where

$$A = \{\langle q, k, \gamma \rangle \mid q \in Q,\ k \in [m],\ \gamma \in \Gamma\}$$

and the transition functions $\tau_{\mathcal{S}}, \tau_{\mathcal{P}}, \tau_1, \ldots, \tau_m$ are defined as follows. Suppose that $a = \langle q, k, \gamma \rangle$ is an element of A. If $\delta(q, \gamma)$ is undefined then the input symbol a is interpreted in each of the automata $\mathcal{S}, \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_m$ as the empty relation, i.e.,

$$\tau_{\mathcal{S}}(a) = \tau_{\mathcal{P}}(a) = \tau_1(a) = \cdots = \tau_m(a) = \emptyset.$$

If $\delta(q, \gamma)$ is defined, say $\delta(q, \gamma) = (r, \gamma', t)$, then

$$\begin{aligned} \tau_{\mathcal{S}}(a) &= \{(q, r)\} \\ \tau_{\mathcal{P}}(a) &= \begin{cases} \{(k, k+t)\} & \text{if } k + t \in [m], \\ \emptyset & \text{if } k + t \notin [m], \end{cases} \\ \tau_i(a) &= \begin{cases} \{(\gamma, \gamma')\} & \text{if } k = i \\ \mathrm{id}_\Gamma & \text{if } k \neq i. \end{cases} \end{aligned}$$

Proof. The intuition is that the automata $\mathcal{S}, \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_m$ together "simulate" the computation of $\mathcal{M}$ on the input word u, such that $\mathcal{S}$ knows the current state of $\mathcal{M}$, $\mathcal{P}$ knows the position of the read-write head, and each $\mathcal{A}_i$ ($i \in [m]$) knows the content of the ith tape-cell. An input symbol $\langle q, k, \gamma \rangle \in A$ corresponds to the statement "the current state of $\mathcal{M}$ is q, the position of the read-write head is k, and the content of the kth tape-cell is γ".

It is easy to see that each one of $\mathcal{S}, \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_m$ is a 1-reset automaton. In order to prove (2.1) consider the product automaton

$$\mathcal{A} = \mathcal{S} \times \mathcal{P} \times \prod_{i \in [m]} \mathcal{A}_i.$$

We know

$$L(\mathcal{A}) = L(\mathcal{S}) \cap L(\mathcal{P}) \cap \bigcap_{i \in [m]} L(A_i).$$

Observe that for all $q, r \in Q$, $v, w \in \Gamma^m$, $j, k \in [m]$, and $a \in A$

$$(q, j, v_1, \ldots, v_m) \xrightarrow{a}_{\mathcal{A}} (r, k, w_1, \ldots, w_m) \iff$$
$$a = \langle q, j, v_j \rangle \wedge \delta(q, v_j) = (r, w_j, k - j) \wedge \forall t \in [m]\,(t \neq j \Rightarrow w_t = v_t),$$

and thus

$$(q, j, vb^\omega) \vdash_{\mathcal{M}} (r, k, wb^\omega) \iff$$
$$\exists a \in A\ (q, j, v_1, \ldots, v_m) \xrightarrow{a}_{\mathcal{A}} (r, k, w_1, \ldots, w_m).$$

It follows that

$$\begin{aligned} u \in L(\mathcal{M}) &\iff (q_0, 1, ub^\omega) \vdash^*_{\mathcal{M}} (q_f, 1, b^\omega) \\ &\iff \exists v \in A^*\ (q_0, 1, (ub^\omega)_1, \ldots, (ub^\omega)_m) \xrightarrow{v}_{\mathcal{A}} (q_f, 1, b, \ldots, b) \\ &\iff L(\mathcal{A}) \neq \emptyset. \ \square \end{aligned}$$

We shall use Construction 2.2.1 to prove that the problem $\mathbf{AIP}_R$ is **PSPACE**-hard. Dexter Kozen [39] used a different construction to prove that the intersection problem of deterministic automata is **PSPACE**-hard. Given a Turing machine $\mathcal{M}$ of space-complexity $p(n)$ and an input word $u \in \Sigma^n$, his construction yields a sequence $\mathcal{D}_1, \ldots, \mathcal{D}_m$ of deterministic automata such that if $\mathcal{M}$ accepts the input word u then the intersection of the languages recognized by these automata consists of the unique word $\#\mathrm{ID}_0\#\mathrm{ID}_1\#\ldots\#\mathrm{ID}_t\#$, where t is the number of computation steps executed by the machine $\mathcal{M}$ on input u, and ID_i $(0 \leq i \leq t)$ is the instantaneous description of the machine $\mathcal{M}$ after executing the ith computation step on input u. If $\mathcal{M}$ rejects u then the intersection is empty. Note that each ID_i is a word of length $p(n) + 1$, and that the automata $\mathcal{D}_1, \ldots, \mathcal{D}_m$ are not aperiodic in general.

Our construction is such that if $\mathcal{M}$ accepts the input word u then the intersection of the languages recognized by the automata $\mathcal{S}, \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_m$ consists of the unique word $\langle q_0, k_0, \gamma_0 \rangle \langle q_1, k_1, \gamma_1 \rangle \ldots \langle q_t, k_t, \gamma_t \rangle$, where q_i is the state of $\mathcal{M}$, k_i is the position of its read-write head, and γ_i is the symbol scanned by the read-write head of $\mathcal{M}$ after executing the ith computation step on input u.

The automata $\mathcal{S}, \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_m$ are aperiodic but not necessarily minimal. In the next construction we modify these automata so that they become minimal. Note that the standard procedure of the minimization of automata is not suitable for our purposes since it requires linear space.

Construction 2.2.2. *Input.* A sequence $\mathcal{A}_1, \ldots, \mathcal{A}_n$ $(n \geq 2)$ of 1-reset automata of the form $\mathcal{A}_i = (Q_i, \Sigma, \tau_i, \{s_i\}, \{f_i\})$.

Output. A sequence $\mathcal{B}_1, \ldots, \mathcal{B}_n$ of minimal 1-reset automata such that

$$\bigcap_{i \in [n]} L(\mathcal{A}_i) = \bigcap_{i \in [n]} L(\mathcal{B}_i). \tag{2.2}$$

Description. For each $i \in [n]$ let

$$\mathcal{B}_i = (Q_i, \Sigma \cup \Sigma', \tau_i', \{s_i\}, \{f_i\}),$$

where

$$\begin{aligned}
\Sigma' &= \{\langle q,j,t\rangle \mid q \in Q_j,\ j \in [n],\ t \in [2]\},\\
\tau_i'(\sigma) &= \tau_i(\sigma),\\
\tau_i'(\langle q,j,1\rangle) &= \begin{cases} \{(s_j,q)\} & \text{if } j=i,\\ \emptyset & \text{if } j\neq i,\end{cases}\\
\tau_i'(\langle q,j,2\rangle) &= \begin{cases} \{(q,f_j)\} & \text{if } j=i,\\ \emptyset & \text{if } j\neq i,\end{cases}
\end{aligned}$$

for all $\sigma \in \Sigma$, $\langle q,j,t\rangle \in \Sigma'$.

Proof. For each $j \in [n]$ let Σ_j' denote the set $\{\langle q,j,t\rangle \mid q \in Q_j,\ t \in [2]\}$. Consider the automaton $\mathcal{B}_i$ for some $i \in [n]$. It is obvious that $\mathcal{B}_i$ is a 1-reset automaton. Since the elements of $\Sigma' \setminus \Sigma_i'$ induce the empty relation in $\mathcal{B}_i$,

$$L(\mathcal{B}_i) \subseteq (\Sigma \cup \Sigma_i')^*.$$

Moreover, since each input symbol $\sigma \in \Sigma$ induces the same relation in $\mathcal{B}_i$ as in $\mathcal{A}_i$,

$$L(\mathcal{B}_i) \cap \Sigma^* = L(\mathcal{A}_i).$$

These two observations and $n \geq 2$ imply (2.2). Lastly, for all states $q,r \in Q_i$ we have $s_i \xrightarrow{\langle q,i,1\rangle}_{\mathcal{B}_i} q \xrightarrow{\langle q,i,2\rangle}_{\mathcal{B}_i} f_i$ and

$$q \neq r \implies r\langle q,i,2\rangle_{\mathcal{B}_i} = \emptyset,$$

showing $\mathcal{B}_i$ is minimal. □

The next construction shows that for each reset automaton $\mathcal{A}$ there exists a "short" regular expression denoting the *complement* of the language recognized by $\mathcal{A}$. This fact plays a key role in proving that the problem $\mathbf{RSF}_R$ is **PSPACE**-hard.

Construction 2.2.3. *Input.* A reset automaton $\mathcal{A} = (Q,\Sigma,\tau,I,F)$.

Output. A regular expression E over Σ such that

$$L(E) = \overline{L(\mathcal{A})}. \tag{2.3}$$

Description. If $I = \emptyset$ then (2.3) holds for the regular expression $E = \Sigma^*$. From now on we assume that $\mathcal{A}$ has an initial state q_0. Let

$$\begin{aligned}
X_q &= \{\sigma \in \Sigma \mid Q\sigma_{\mathcal{A}} = \{q\}\}\\
Y_q &= \{\sigma \in \Sigma \mid q\sigma_{\mathcal{A}} = \{q\}\}\\
Z_q &= \{\sigma \in \Sigma \mid q\sigma_{\mathcal{A}} = \emptyset\},
\end{aligned}$$

for all $q \in Q$. Using these subsets of Σ we define the regular expressions

$$E_q = \begin{cases} \Sigma^* X_q Y_q^* & \text{if } q \neq q_0\\ \Sigma^* X_q Y_q^* \cup Y_q^* & \text{if } q = q_0,\end{cases}$$

for all $q \in Q$. Lastly, let

$$E = \left(\bigcup_{q \in Q \setminus F} E_q \right) \cup \left(\bigcup_{q \in Q} E_q Z_q \Sigma^* \right).$$

Proof. We claim

$$u \in L(E_q) \implies q_0 u_{\mathcal{A}} \subseteq \{q\} \tag{2.4}$$

and

$$q_0 u_{\mathcal{A}} = \{q\} \implies u \in L(E_q), \tag{2.5}$$

for all $q \in Q$, $u \in \Sigma^*$. Then (2.3) follows since the definition of E expresses the fact that an input word $u \in \Sigma^*$ is rejected by the automaton $\mathcal{A}$ if either $q_0 u_{\mathcal{A}} = \{q\}$ for some non-final state q, or $q_0 u_{\mathcal{A}} = \emptyset$.

As (2.4) is quite obvious, we only prove (2.5). Suppose that $q_0 u_{\mathcal{A}} = \{q\}$ for some state $q \in Q$ and input word $u \in \Sigma^n$, $n \geq 0$. Then there exist some states $q_1, \ldots, q_{n-1}$ such that there is a directed walk

$$q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} q_{n-1} \xrightarrow{u_n} q$$

in $\mathcal{A}$. If $q_0 = q_1 = \cdots = q_{n-1} = q$ then $u \in Y_q^* \subseteq L(E_q)$. Otherwise let $k \in [n]$ be the largest index for which $q_{k-1} \neq q$. Then we have

$$q \neq q_{k-1} \xrightarrow{u_k} q \xrightarrow{u_{k+1}} \cdots \xrightarrow{u_{n-1}} q \xrightarrow{u_n} q.$$

Since $\mathcal{A}$ is a deterministic automaton it follows that $u_{k+1}, \ldots, u_n \in Y_q$. Moreover, since $q \neq q_{k-1} \xrightarrow{u_k} q$, the relation induced by u_k is not the identity function. Thus, u_k induces a partial constant function with range $\{q\}$, so that $u_k \in X_q$. It follows that $u \in \Sigma^* X_q Y_q^* \subseteq L(E_q)$. □

The next construction presents the main idea of reducing $\mathbf{AIP}_R$ to $\mathbf{ASF}_R$. The very same idea was used by Cho and Huynh in [20].

Construction 2.2.4. *Input.* A sequence $\mathcal{B}_1, \ldots, \mathcal{B}_n$ ($n \geq 2$) of minimal 1-reset automata of the form $\mathcal{B}_i = (Q_i, \Sigma, \tau_i, \{s_i\}, \{f_i\})$.

Output. A minimal DFA $\mathcal{C}$ such that

$$\bigcap_{i \in [n]} L(\mathcal{B}_i) = \emptyset \iff L(\mathcal{C}) \text{ is star-free.} \tag{2.6}$$

Description. Observe that $L(B_i) \neq \emptyset$, for all $i \in [n]$. Let p be the least prime number with $p \geq n$. Note that $p < 2n$ (for a proof see [37]), and thus the trivial algorithm can find p using only logarithmic space. For integers $i \in \{n+1, n+2, \ldots, p\}$ let $\mathcal{B}_i = (Q_i, \Sigma, \tau_i, \{s_i\}, \{f_i\})$ be the minimal 1-reset automaton recognizing the language Σ^*. For the sake of simplicity assume that the sets Q_i ($i \in [p]$) are pairwise disjoint,

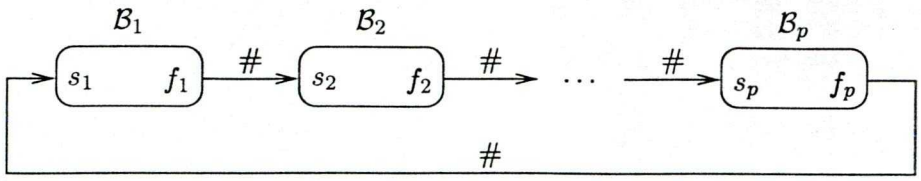


Figure 2.1: The automaton $\mathcal{C}$

and that $\# \notin \Sigma$ is a new input symbol. Let $\nu : \mathbf{N} \to [p]$ be the function mapping each integer i to $((i-1) \bmod p)+1$. Then we define $\mathcal{C} = (\bigcup_{i\in[p]} Q_i, \Sigma \cup \{\#\}, \tau, \{s_1\}, \{s_1\})$, where

$$\begin{aligned} \tau(\#) &= \{(f_i, s_{\nu(i+1)}) \mid i \in [p]\} \\ \tau(\sigma) &= \bigcup_{i\in[p]} \tau_i(\sigma), \end{aligned}$$

for all input symbols $\sigma \in \Sigma$. See Figure 2.1.

Proof. Clearly, $\mathcal{C}$ is a DFA with

$$L(\mathcal{C}) = \left(L(\mathcal{B}_1)\#L(\mathcal{B}_2)\#\cdots L(\mathcal{B}_n)\#(\Sigma^*\#)^{p-n}\right)^*.$$

By Schützenberger's theorem, (2.6) is equivalent to the condition

$$\bigcap_{i\in[n]} L(\mathcal{B}_i) \neq \emptyset \iff \mathcal{C} \text{ is not aperiodic.} \tag{2.7}$$

The "$\Longrightarrow$" part of (2.7) is obvious. If $u \in \Sigma^*$ is a common element of the languages $L(B_1), \ldots, L(\mathcal{B}_n)$ then $s_1 \xrightarrow{(u\#)^p}_{\mathcal{C}} s_1$ and $s_1 \xrightarrow{u\#}_{\mathcal{C}} s_2 \neq s_1$, so that $\mathcal{C}$ is not aperiodic by Remark 1.4.1. As a first step for proving the "$\Longleftarrow$" part of (2.7) observe that if the letter # appears l times in an input word $u \in (\Sigma \cup \{\#\})^*$ and $q \in Q_i$ is a state such that $q(u\#)_{\mathcal{C}} \neq \emptyset$, then $q(u\#)_{\mathcal{C}} = \{s_{\nu(i+l+1)}\}$. Moreover, if $s_j(v\#)_{\mathcal{C}} \neq \emptyset$ for some integer $j \in [p]$ and word $v \in \Sigma^*$ then $v \in L(\mathcal{B}_j)$. Now suppose that $\mathcal{C}$ is not aperiodic, i.e.

$$q \xrightarrow{u^k}_{\mathcal{C}} q, \tag{2.8}$$

and

$$q \xrightarrow{u}_{\mathcal{C}} q', \tag{2.9}$$

for some *different* states $q \in Q_i$, $q' \in Q_{i'}$, $i, i' \in [p]$, input word $u \in (\Sigma \cup \{\#\})^+$ and integer $k \geq 2$. Note that by (2.8) we have $q(u^t)_{\mathcal{C}} \neq \emptyset$, for all $t \geq 0$. Let l be the number of #'s in u, so that u can be written as

$$u^{(0)}\#u^{(1)}\#\ldots\#u^{(l)},$$

where $u^{(0)}, \ldots, u^{(l)}$ are words in Σ^*. If l were 0 then we would have $i' = i$, $q \xrightarrow{u^k}_{\mathcal{B}_i} q$, and $q \xrightarrow{u}_{\mathcal{B}_i} q' \neq q$, contradicting the fact that $\mathcal{B}_i$ is aperiodic. Thus, $l > 0$.

Let v denote the word $u^{(0)}\# \cdots \# u^{(l-1)}$, so that $u = v\#u^{(l)}$ and

$$q \xrightarrow{v\#}_{\mathcal{C}} s_j,$$

where $j = \nu(i + l)$. By (2.8) we have

$$q \xrightarrow{u^{k-1}v\#}_{\mathcal{C}} s_i \xrightarrow{u^{(l)}}_{\mathcal{C}} q.$$

If p were a divisor of l then it would follow that $j = i$ and $q \xrightarrow{v\#}_{\mathcal{C}} s_i \xrightarrow{u^{(l)}}_{\mathcal{C}} q$, contradicting (2.9). Thus p is not a divisor of l.

Let j be an arbitrary element of $[n]$. As p is a prime not dividing l, there exists some integer $t \geq 0$ such that $\nu(i + lt) = j$. For this t we have

$$q \xrightarrow{u^{t-1}v\#}_{\mathcal{C}} s_j.$$

Moreover, since $u^{t-1}v\#u^{(l)}u^{(0)}\#$ is a prefix of u^{t+1} and $q(u^{t+1})_{\mathcal{C}} \neq \emptyset$, it follows that

$$s_j(u^{(l)}u^{(0)}\#)_{\mathcal{C}} = q(u^{t-1}v\#u^{(l)}u^{(0)}\#)_{\mathcal{C}} \neq \emptyset,$$

showing $u^{(l)}u^{(0)} \in L(\mathcal{B}_j)$. Since $j \in [n]$ was arbitrary,

$$u^{(l)}u^{(0)} \in \bigcap_{j \in [n]} L(\mathcal{B}_j).$$

In order to prove $\mathcal{C}$ is minimal suppose that $q \in Q_j$ and $r \in Q_k$ are two different states of the automaton $\mathcal{C}$. For each $i \in [p]$ choose an arbitrary word $v^{(i)} \in L(\mathcal{B}_i)$. Since q is a biaccessible state of $\mathcal{B}_j$, there exist words $v, w \in \Sigma^*$ with

$$s_j \xrightarrow{v}_{\mathcal{B}_j} q \xrightarrow{w}_{\mathcal{B}_j} f_j.$$

Then

$$s_1 \xrightarrow{v^{(1)}\#\cdots\#v^{(j-1)}\#v}_{\mathcal{C}} q \xrightarrow{w\#v^{(j+1)}\#\cdots\#v^{(p)}\#}_{\mathcal{C}} s_1,$$

showing q is a biaccessible state of $\mathcal{C}$. If $j \neq k$, say $j < k$, then

$$\begin{aligned} q(w\#v^{(j+1)}\#\cdots\#v^{(p)}\#)_{\mathcal{C}} &= \{s_1\} \quad \text{and} \\ r(w\#v^{(j+1)}\#\cdots\#v^{(p)}\#)_{\mathcal{C}} &\subseteq \{s_{k-j+1}\}. \end{aligned}$$

Lastly, suppose that $j = k$. Since $\mathcal{B}_j$ is minimal, there exists some word $x \in \Sigma^*$ such that exactly one of the sets $qx_{\mathcal{B}_j} \cap \{f_j\}$ and $rx_{\mathcal{B}_j} \cap \{f_j\}$ is empty. We may assume $f_j \in qx_{\mathcal{B}_j}$ and $f_j \notin rx_{\mathcal{B}_j} \cap \{f_j\}$. Then $q(x\#)_{\mathcal{C}} = \{s_{\nu(j+1)}\}$, $r(x\#)_{\mathcal{C}} = \emptyset$ and we have

$$\begin{aligned} q(x\#v^{(j+1)}\#\cdots\#v^{(p)}\#)_{\mathcal{C}} &= \{s_1\} \quad \text{and} \\ r(x\#v^{(j+1)}\#\cdots\#v^{(p)}\#)_{\mathcal{C}} &= \emptyset. \ \square \end{aligned}$$

We say that a function $\varphi : A^* \to B^*$ is a *word-homomorphism* if $\varphi(\epsilon) = \epsilon$ and $\varphi(uv) = \varphi(u)\varphi(v)$, for all words $u, v \in A^*$. Thus, a word-homomorphism is just a homomorphism from the free monoid $(A^*; \cdot, \epsilon)$ into the free monoid $(B^*; \cdot, \epsilon)$. Note that each word-homomorphism $A^* \to B^*$ is totally determined by its restriction to A.

For the next construction we need the following simple (but rather technical) lemma. This lemma is also needed for proving Theorem 2.3.4.

Lemma 2.2.1. *Suppose that* $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ *and* $l = \lceil \log_2(n+1) \rceil$. *Then there exists an injective word-homomorphism* $\psi : \Sigma^* \to \{0,1\}^*$ *satisfying the following conditions.*

- *The* ψ*-image of each letter* $\sigma \in \Sigma$ *is a word of length* $2l$ *beginning with a sequence of* l *zeros and containing the letter* 1. *In other words,*

$$\psi(\Sigma) \subseteq 0^l\{0,1\}^l \setminus \{0^{2l}\}. \tag{2.10}$$

- *For all words* $u, v, w \in \{0,1\}^*$

$$uv^{3l-1}w \in \psi(\Sigma^*) \implies 2l \text{ divides } |v|. \tag{2.11}$$

- *For all languages* $L \subseteq \Sigma^*$

$$L \text{ is star-free} \iff \psi(L) \text{ is star-free.} \tag{2.12}$$

Proof. First of all note that $n \leq 2^l - 1$, so that l bits are sufficient to represent the number n in binary. Let $\psi : \Sigma^* \to \{0,1\}^*$ be the word-homomorphism mapping each letter $\sigma_i \in \Sigma$ ($i \in [n]$) to the $2l$-bit binary representation of i. Then ψ is injective and satisfies (2.10).

Suppose that (2.11) is not true. Then there exist some words $u, v, w \in \{0,1\}^*$ such that $uv^{3l-1}w \in \psi(\Sigma^*)$, and $2l$ is not a divisor of $|v|$. Let us denote $|v|$ by m. Then $m > 0$ and $\gcd(2l, m) < 2l$. Since none of the integers $l+1, l+2, \ldots, 2l-1$ is a divisor of $2l$,

$$\gcd(2l, m) \leq l. \tag{2.13}$$

Moreover, since no word in $\psi(\Sigma^*)$ may contain 0^{3l-1} as a subword (the longest possible sequence of zeros is of length $3l-2$; it appears in the the word $\psi(\sigma_{2^{l-1}}\sigma_1) = 0^l 10^{l-1}0^{2l-1}1$), the letter 1 occurs in the word v^{3l-1}, and thus in v. Let $j \in [m]$ be an integer such that the jth letter of v is 1. If $i \in [2lm]$ is an integer satisfying

$$i \equiv j \pmod{m} \tag{2.14}$$

then the ith letter of v^{2l} is 1. By (2.10) it follows that if $1 \leq i \leq |uv^{2l}w|$ is an integer such that $(i-1) \bmod 2l < l$, then the ith letter of $uv^{2l}w$ is 0. Thus, if $i \in [2lm]$ is an integer satisfying

$$i \equiv t - |u| \pmod{2l} \tag{2.15}$$

for some $t \in [l]$, then the ith letter of v^{2l} is 0. The diophantic system (2.14,2.15) is solvable in the variable i if and only if

$$t - |u| \equiv j \pmod{\gcd(2l, m)}, \tag{2.16}$$

and in this case each solution i can be written in the form

$$i = i_0 + h \cdot \operatorname{lcm}(2l, m),$$

where i_0 is a fixed solution and h is an integer. Let t be the unique element of $[\gcd(2l, m)]$ satisfying (2.16). Then $t \in [l]$, by (2.13). For this t there exists a unique integer $i \in [\operatorname{lcm}(2l, m)] \subseteq [2lm]$ such that both (2.14) and (2.15) hold. But then we have the contradiction that the ith letter of v^{2l} is equal to both 0 and 1. This contradiction was caused by the assumption that (2.11) fails.

In order to prove (2.12) suppose that $L \subseteq \Sigma^*$ is a language and $\psi(L) \subseteq \{0,1\}^*$ is star-free. Then L is regular and there exists an integer $k \geq 0$ such that for all words $u, v, w \in \Sigma^*$,

$$\begin{aligned} uv^k w \in L &\iff \psi(u)\psi(v)^k\psi(w) \in \psi(L) \\ &\iff \psi(u)\psi(v)^{k+1}\psi(w) \in \psi(L) \\ &\iff uv^{k+1}w \in L, \end{aligned}$$

showing L is star-free. Thus, for this direction no special property of the word-homomorphism ψ is needed other than its injectivity.

For the converse direction, suppose that $L \subseteq \Sigma^*$ is a star-free language. Then $\psi(L)$ is regular, and there exists an integer $k \geq 0$ such that

$$xy^k z \in L \iff xy^{k+1}z \in L, \tag{2.17}$$

for all words $x, y, z \in \Sigma^*$. Let m be the maximum of $3l - 1$ and $k + 1$. Suppose that $uv^m w \in \psi(L)$, for some $u, v, w \in \{0,1\}^*$. We want to show that $uv^{m+1}w \in \psi(L)$. This is obvious if $|v| = 0$, so suppose that $|v| > 0$. By (2.11) it follows that $|v|$ is a multiple of $2l$, so that $|v| \geq 2l$. Let α be the shortest prefix of v such that the length of the word $u\alpha$ is a multiple of $2l$. Then v can be written as $\alpha\beta$, for some word $\beta \in \{0,1\}^*$. Since $uv^m w = u\alpha(\beta\alpha)^{m-1}\beta w \in \psi(L)$ and the length of the words $u\alpha$, $\beta\alpha$ and βw are multiples of $2l$, there exist words $x, y, z \in \Sigma^*$ such that $\psi(x) = u\alpha$, $\psi(y) = \beta\alpha$, $\psi(z) = \beta w$, and $xy^{m-1}z \in L$. Since $m - 1 \geq k$, it follows by (2.17) that $xy^m z \in L$. Thus,

$$\psi(xy^m z) = u\alpha(\beta\alpha)^m\beta w = uv^{m+1}w \in \psi(L).$$

The implication $uv^{m+1}w \in \psi(L) \Longrightarrow uv^m w \in \psi(L)$ is proved in a similar way. □

The last construction gives the second part of the reduction $\mathbf{AIP}_R \leq_{log} \mathbf{ASF}_R$.

Construction 2.2.5. *Input.* A minimal DFA $C = (Q, \Sigma, \tau, I, F)$.

Output. A minimal DFA $\mathcal{C}'$ with input symbols $\{0,1\}$ such that

$$L(\mathcal{C}) \text{ is star-free} \iff L(\mathcal{C}') \text{ is star-free.} \tag{2.18}$$

Description. Let $\psi : \Sigma^* \to \{0,1\}^*$ be an injective word-homomorphism satisfying the conditions of Lemma 2.2.1. In particular, the image $\psi(\sigma)$ of each symbol $\sigma \in \Sigma$ is a word in $\{0,1\}^{2l}$, where $l = \lceil \log_2(|\Sigma| + 1) \rceil$. The idea of the construction is that $\mathcal{C}'$ should be a minimal DFA recognizing the language $\psi(L(\mathcal{C}))$. We give the concrete description of one such automaton $\mathcal{C}'$.

For each state $q \in Q$ let

$$S_q := \{\psi(\sigma)q' \mid \sigma \in \Sigma,\ q' \in Q,\ q \xrightarrow{\sigma}_{\mathcal{C}} q'\},$$

so that S_q is a set of words with letters in $\{0,1\} \cup Q$, more precisely, $S_q \subseteq \{0,1\}^{2l}Q$. When S is a set of words and u is a word, $u\backslash S$ denotes the set $\{v \mid uv \in S\}$. For each integer $j \in [2l-1]$ let

$$Q'_j := \{u\backslash S_q \mid q \in Q,\ u \in \{0,1\}^j\} \setminus \{\emptyset\}.$$

Thus each element of Q'_j is a nonempty subset of $\{0,1\}^{2l-j}Q$. Now let

$$\mathcal{C}' := (Q \cup Q', \{0,1\}, \tau', I, F),$$

where

$$Q' = \bigcup_{j \in [2l-1]} Q'_j,$$

and τ' is defined such that

$$qx_{\mathcal{C}'} = \begin{cases} \{x\backslash S_q\} & \text{if } x\backslash S_q \neq \emptyset, \\ \emptyset & \text{otherwise,} \end{cases}$$

$$Sx_{\mathcal{C}'} = \begin{cases} \{q'\} & \text{if } x\backslash S = \{q'\}, \text{ for some } q' \in Q, \\ \emptyset & \text{if } x\backslash S = \emptyset, \\ \{x\backslash S\} & \text{otherwise,} \end{cases}$$

for all $q \in Q$, $S \in Q'$, $x \in \{0,1\}$.

Proof. Let us denote Q by Q'_0. It is easy to see that $\mathcal{C}'$ is a DFA satisfying

$$q \xrightarrow{u}_{\mathcal{C}'} q' \iff \exists v \in \Sigma^*\ u = \psi(v) \wedge q \xrightarrow{v}_{\mathcal{C}} q', \tag{2.19}$$

and

$$Q'_i \xrightarrow{u}_{\mathcal{C}'} Q'_j \implies |u| \equiv j - i \pmod{2l}, \tag{2.20}$$

for all $q, q' \in Q$, $u \in \{0,1\}^*$, $0 \leq i, j < 2l$. It follows in particular that $L(\mathcal{C}') = \psi(L(\mathcal{C}))$, so that (2.18) holds by Lemma 2.2.1.

In order to prove C' is minimal suppose that $s \in Q'_i$ and $s' \in Q'_j$ $(0 \leq i, j < 2l)$ are two different states of C'. It is clear from the description of C' that there exist words $v \in \{0,1\}^i$, $v' \in \{0,1\}^{2l-i}$, and states $q, q' \in Q$ such that $q \xrightarrow{v}_{C'} s \xrightarrow{v'}_{C'} q'$. Since C is minimal, there exist words $u, u' \in \Sigma^*$ with $I \xrightarrow{u}_C q$ and $q' \xrightarrow{u'}_C F$. By (2.19) we have

$$I \xrightarrow{\psi(u)}_{C'} q \xrightarrow{v}_{C'} s \xrightarrow{v'}_{C'} q' \xrightarrow{\psi(u')}_{C'} F,$$

showing s is a biaccessible state of C'. If $i \neq j$ then s and s' are not equivalent by (2.20), so suppose $i = j$. If $i = j = 0$ then s and s' are two different elements of Q, and since C is minimal there exists a word $w \in \Sigma^*$ such that exactly one of the two sets $sw_C \cap F = s\psi(w)_{C'} \cap F$ and $s'w_C \cap F = s'\psi(w)_{C'} \cap F$ is empty. Lastly suppose that $i = j \in [2l-1]$. Then s and s' are two different subsets of the set $\{0,1\}^{2l-i}Q$, say $s \not\subseteq s'$. Let uq be an arbitrary element in s which is not in s', where $u \in \{0,1\}^{2l-i}$ and $q \in Q$. There are two possibilities: either $s'u_{C'} = \emptyset$ or $s'u_{C'} = \{q'\}$ for some state $q' \in Q$, $q' \neq q$. In the first case we have $su\psi(v)_{C'} \cap F \neq \emptyset$ and $s'u\psi(v)_{C'} \cap F = \emptyset$, where $v \in \Sigma^*$ is an arbitrary word with $q \xrightarrow{v}_C F$. (Such a v exists since q is a coaccessible state of C.) The second case can be handled similarly to the case $i = j = 0$. □

2.3 Results

Theorem 2.3.1. *The problems* **AIP** *and* $\mathbf{AIP}_R$ *are* **PSPACE***-complete with respect to logspace reductions.*

Proof. We show

$$\mathbf{PSPACE} \leq_{log} \mathbf{AIP}_R \leq_{log} \mathbf{AIP} \in \mathbf{PSPACE}.$$

Suppose that $L \subseteq \Sigma^*$ is a language in **PSPACE**. Then there exists a polynomial function $p : \mathbf{N} \to \mathbf{N}$ and a deterministic Turing machine $\mathcal{M}$ of space-complexity p such that $L(\mathcal{M}) = L$. Applying Construction 2.2.1 followed by Construction 2.2.2 to $\mathcal{M}$ and an input word $u \in \Sigma^*$, we obtain a list $\mathcal{A}_1, \ldots, \mathcal{A}_n$ of minimal 1-reset automata such that

$$u \in L \iff \bigcap_{i \in [n]} L(\mathcal{A}_i) \neq \emptyset.$$

Since both constructions can be carried out by a logspace-bounded Turing machine, $\mathbf{PSPACE} \leq_{log} \mathbf{AIP}_R$. The claim $\mathbf{AIP}_R \leq_{log} \mathbf{AIP}$ is trivial.

Kozen proved in [39] that the intersection problem of deterministic finite automata is decidable in nondeterministic linear space, and thus it is in **PSPACE**by Savitch's theorem. We do the same for nondeterministic automata, that is, we prove that $\mathbf{AIP} \in \mathbf{PSPACE}$.

Suppose that $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are NFA's with input symbols Σ, say $\mathcal{A}_i = (Q_i, \Sigma, \tau_i, I_i, F_i)$. It is easy to see that the following nondeterministic PASCAL-style program accepts the automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ if and only if $\bigcap_{i \in [n]} L(\mathcal{A}_i) \neq \emptyset$:

```
function Solve_AIP(A_1,... A_n : NFA):boolean;
var
    S_1,... ,S_n : set of state;
    σ : input symbol;
begin
    S_1 := I_1;
        ⋮
    S_n := I_n;
    while S_1 ∩ F_1 = ∅ or ··· or S_n ∩ F_n = ∅ do
    begin
        guess σ ∈ Σ;
        S_1 := S_1 σ_{A_1};
            ⋮
        S_n := S_n σ_{A_n};
    end;
    Solve_AIP:=true;
end;
```

The idea is simple: we simulate all possible computations of the product automaton $\prod_{i\in[n]} \mathcal{A}_i$ on a random input word in parallel until a final state is reached. The space complexity of the program is linear. It follows by Savitch's theorem [46] that **AIP** $\in$ **PSPACE**. □

Theorem 2.3.2. *The problem* $\mathbf{AIP}_C$ *is solvable in polynomial time.*

Proof. Suppose that $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are complete reset automata with input symbols in Σ, say

$$\mathcal{A}_i = (Q_i, \Sigma, \tau_i, \{s_i\}, F_i),$$

where

$$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\},$$

for some $m \geq 0$. The following deterministic PASCAL-style program accepts these automata (i.e., returns `true`) if and only if the language

$$L = \bigcap_{i\in[n]} L(\mathcal{A}_i)$$

is not empty, and in this case it also returns a word $u \in L$. Intuitively, the program builds up the word u letter by letter in reverse direction, it stores the current postfix of u in the variable w, and it also keeps track of those automata in which w induces the identity function.

```
function Solve_AIP_C(A_1,... A_n :NFA; var u :input word):boolean;
var
```

```
    w : input word;
    I, J : set of integer;
    reject : boolean;
begin
    w := ε;
    I := [n];
    if each A_i (1 ≤ i ≤ n) is a complete reset automaton then
        reject := false
    else
        reject := true;
    while not reject and ∃i ∈ I s_i ∉ F_i do
    begin
        J := {j ∈ [m] | ∀i ∈ I (τ_i(σ_j) = id_{Q_i} or Rng(τ_i(σ_j)) ⊆ F_i)};
        J := J \ {j ∈ [m] | ∀i ∈ I τ_i(σ_j) = id_{Q_i}};
        if J = ∅ then
            reject := true;
        else
            begin
                w := σ_{min J} · w;
                I := I \ {i ∈ I | Rng(τ_i(σ_k)) ⊆ F_i};
            end;
    end;
    u := w;
    Solve_AIP_C := not reject;
end;
```

The correctness of the program follows easily once we have proved that at the beginning of each execution of the `while` loop the following conditions hold:

$$L \neq \emptyset \implies \exists v \in \Sigma^* \; vw \in L \tag{2.21}$$

$$reject \implies \forall v \in \Sigma^* \; vw \notin L \tag{2.22}$$

$$\forall i \in [n] \quad i \in I \iff w_{\mathcal{A}_i} = \mathrm{id}_{Q_i} \tag{2.23}$$

$$\forall i \in [n] \quad i \notin I \implies \mathrm{Rng}(w_{\mathcal{A}_i}) \subseteq F_i. \tag{2.24}$$

These conditions clearly hold when the program first enters the loop. Now suppose that *reject* is false, $s_i \notin F_i$ for some $i \in [n]$, and the above conditions hold. Then it follows that $w \notin L$, which together with (2.21) implies that if $L \neq \emptyset$ then there exists a *nonempty* word $v \in \Sigma^+$ such that $vw \in L$. The last letter of such a word v does not induce a constant function mapping each state to a nonfinal state in any of the automata $\mathcal{A}_i$, $i \in I$, because otherwise it would follow by (2.23) that at least one of these automata does not accept the word vw. Thus, the first instruction of the loop's core collects in J the indices of those input symbols which are candidates for being the last letter of v. How do we know which of these candidates is the right one? They are equally good in the sense that if there exists a word v with $vw \in L$ then for all $j \in J$ there exists a word v' ending in σ_j such that $v'w \in L$. Indeed, if $vw \in L$ and $j \in J$, then it follows by (2.24) that the word $v\sigma_j w$ also belongs to L.

On the other hand, if $j \in J$, $v\sigma_j w \in L$ and σ_j induces the identity function in each of the automata $\mathcal{A}_i$, $i \in I$, then it follows again by (2.24) that $vw \in L$. This means that after executing the second instruction of the loop we know that if there is a word $v \in \Sigma^*$ such that $vw \in L$ then for all $j \in J$ there exists a word $v' \in \Sigma^*$ with $v'\sigma_j w \in L$. Thus if $J = \emptyset$ then there is no word v with $vw \in L$, otherwise we may choose any symbol σ_j with $j \in J$ as the last letter of v. Now it is easy to see that the conditions (2.21–2.24) will hold after the execution of the `while` loop's core.

The loop itself terminates either if *reject* becomes `true`, or if *reject* is `false` and $s_i \in F_i$ for all $i \in I$. In the first case it follows by (2.21) and (2.22) that $L = \emptyset$, while in the second case conditions (2.23) and (2.24) imply $w \in L$.

As for the complexity of the algorithm, it can be decided in polynomial time if each $\mathcal{A}_i$ is a complete reset automaton. Moreover, observe that in each execution of the `while` loop's core the cardinality of I decreases by at least 1. Since at the beginning $|I| = n$, the loop will terminate after at most n executions. □

Theorem 2.3.3. *The problems* **ASF** *and* $\mathbf{ASF}_R$ *are* **PSPACE***-complete with respect to logspace reductions.*

Proof. We show

$$\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{ASF}_R \leq_{log} \mathbf{ASF} \in \mathbf{PSPACE},$$

where $\overline{\mathbf{AIP}_R}$ is the complement of $\mathbf{AIP}_R$, that is, the problem of deciding if the intersection of the languages recognized by some given minimal 1-reset automata *is* empty.

Suppose that $\mathcal{B}_1, \ldots, \mathcal{B}_n$ ($n \geq 2$) are minimal 1-reset automata with a common set of input symbols. Applying Construction 2.2.4 to $\mathcal{B}_1, \ldots, \mathcal{B}_n$, followed by Construction 2.2.5, we obtain a minimal DFA $\mathcal{C}'$ with input symbols $\{0, 1\}$ such that

$$\bigcap_{i \in [n]} L(\mathcal{B}_i) = \emptyset \iff L(\mathcal{C}') \text{ is star-free.}$$

Since both constructions can be carried out by a logspace-bounded Turing machine, $\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{ASF}_R$. The claim $\mathbf{ASF}_R \leq_{log} \mathbf{ASF}$ is trivial, so it remains to show that the problem **ASF** is decidable in polynomial space. Suppose that $\mathcal{A} = (Q, \Sigma, \tau, I, F)$ is an NFA. The idea is that we apply Stern's algorithm [49] to the minimal DFA recognizing the language $L(\mathcal{A})$. We must do this whithout actually constructing the minimal DFA, since it can be exponentially large compared to $\mathcal{A}$. Recall that the power automaton of $\mathcal{A}$ is the deterministic automaton

$$\mathrm{P}(\mathcal{A}) = (\mathrm{P}(Q), \Sigma, \tau', \{I\}, F'),$$

where

$$\begin{aligned} F' &= \{S \in \mathrm{P}(Q) \mid S \cap F \neq \emptyset\}, \\ \tau'(\sigma) &= \{(S, S\sigma_{\mathcal{A}}) \mid S \in \mathrm{P}(Q)\}, \end{aligned}$$

for all $\sigma \in \Sigma$. The minimal DFA recognizing $L(\mathcal{A})$ is obtained from $\mathrm{P}(\mathcal{A})$ by deleting those states which are not biaccessible, and then identifying the equivalent states. It follows that $L(\mathcal{A})$ is *not* star-free if and only if there exists some input word $u \in \Sigma^*$, *accessible* state S of $\mathrm{P}(\mathcal{A})$ and integer $k \geq 2$ such that $S \approx_{\mathrm{P}(\mathcal{A})} S(u^k)_{\mathcal{A}} = S(u_{\mathcal{A}})^k$ and $S \not\approx_{\mathrm{P}(\mathcal{A})} Su_{\mathcal{A}}$. The following nondeterministic procedure decides if $S \not\approx_{\mathrm{P}(\mathcal{A})} S'$ holds for two states S, S' of $\mathrm{P}(\mathcal{A})$:

```
function Not_Equiv(S, S' : set of state):boolean;
var
    σ : input symbol;
begin
    while (S ∩ F = ∅ and S' ∩ F = ∅) or
          (S ∩ F ≠ ∅ and S' ∩ F ≠ ∅) do
    begin
        guess σ ∈ Σ;
        S := Sσ_A;
        S' := S'σ_A;
    end;
    Not_Equiv:=true;
end;
```

By Savitch's theorem we obtain a deterministic program `Equiv` of polynomial space-complexity which decides if two states of $\mathrm{P}(\mathcal{A})$ are equivalent. The following nondeterministic program uses `Equiv` as a subroutine to decide if $L(\mathcal{A})$ is not star-free:

```
function Not_ASF(A : NFA):boolean;
var
    σ : input symbol;
    S, S' : set of state;
    ρ : relation;
    halt : boolean;
begin
    S := I;
    repeat
        guess σ ∈ Σ;
        S := Sσ_A;
        guess halt;
    until halt;
    ρ := ε_A;
    repeat
        guess σ ∈ Σ;
        ρ := ρ ∘ σ_A;
        guess halt;
    until halt;
    S' := Sρ;
    if Equiv(S, S') then
```

```
        Not_ASF:=false
    else begin
        repeat
            S' := S'ρ;
        until Equiv(S, S');
        Not_ASF:=true;
    end;
end;
```

By Savitch's theorem and the fact that the language class **PSPACE** is closed under complementation it follows that **ASF** $\in$ **PSPACE**. $\square$

Theorem 2.3.4. *The problems* **RSF** *and* $\mathbf{RSF}_R$ *are* **PSPACE***-complete with respect to logspace reductions.*

Proof. We show

$$\overline{\mathbf{AIP}_R} \leq_{log} \mathbf{RSF}_R \leq_{log} \mathbf{RSF} \leq_{log} \mathbf{ASF}.$$

The claim $\mathbf{RSF}_R \leq_{log} \mathbf{RSF}$ is trivial, and it is also easy to see that $\mathbf{RSF} \leq_{log} \mathbf{ASF}$: given a regular expression E, a logspace-bounded Turing machine can construct a *nondeterministic* automaton $\mathcal{A}$ such that $L(E) = L(\mathcal{A})$.

Suppose that $\mathcal{B}_1, \ldots, \mathcal{B}_n$ $(n \geq 2)$ are minimal 1-reset automata with input symbols in Σ. Let $\mathcal{C}$ be the result of Construction 2.2.4 applied to the automata $\mathcal{B}_1, \ldots, \mathcal{B}_n$. Then $\mathcal{C}$ is a minimal DFA with input symbols in $\Sigma \cup \{\#\}$ such that

$$\bigcap_{i \in [n]} L(\mathcal{B}_i) = \emptyset \iff L(\mathcal{C}) \text{ is star-free.}$$

Applying Construction 2.2.3 to each one of the automata $\mathcal{B}_1, \ldots, \mathcal{B}_n$ we get regular expressions $E_1, \ldots, E_n$ such that

$$L(E_i) = \overline{L(\mathcal{B}_i)},$$

for all $i \in [n]$. Recall that

$$L(\mathcal{C}) = \left(L(\mathcal{B}_1)\#L(\mathcal{B}_2)\# \cdots L(\mathcal{B}_n)\#(\Sigma^*\#)^{p-n}\right)^*,$$

where p is the smallest prime number such that $p \geq n$. It follows that a word $v = v^{(0)}\#v^{(1)}\# \cdots v^{(k-1)}\#v^{(k)}$ $(k \geq 0,\ v^{(0)}, \ldots, v^{(k)} \in \Sigma^*)$ belongs to $L(\mathcal{C})$ if and only if $v^{(k)} = \epsilon$, k is a multiple of p, and $v^{(i)} \in L(\mathcal{B}_{(i \bmod p)+1})$, for all $i < k$ with $i \bmod p < n$. The languages denoted by the regular expressions

$$\begin{aligned}
F_1 &= (\Sigma \cup \#)^*\Sigma \\
F_2 &= ((\Sigma^*\#)^p)^* \left(\bigcup_{i \in [p-1]} (\Sigma^*\#)^i \right) \Sigma^* \\
F_3 &= ((\Sigma^*\#)^p)^* \left(\bigcup_{i \in [n]} (\Sigma^*\#)^{i-1} E_i \# \right) (\Sigma \cup \#)^*
\end{aligned}$$

consist of those words $v = v^{(0)}\#v^{(1)}\#\cdots v^{(k-1)}\#v^{(k)}$ for which

1. $v^{(k)} \neq \epsilon$,
2. k is not a multiple of p,
3. $v^{(i)} \notin L(B_{(i \bmod p)+1})$ for some $i < k$ with $i \bmod p < n$,

respectively. Thus, the regular expression $E := F_1 \cup F_2 \cup F_3$ denotes the complement of the language $L(\mathcal{C})$. Let $\psi : (\Sigma \cup \{\#\})^* \to \{0,1\}^*$ be a word-homomorphism satisfying the conditions of Lemma 2.2.1. Let E' be the regular expression obtained from E by replacing each occurrence of every letter $x \in \Sigma \cup \{\#\}$ by the word $\psi(x) \in \{0,1\}^*$. Then E' is a regular expression over $\{0,1\}$ having star-height 2. Moreover, $L(E') = \psi(L(E)) = \psi(\overline{L(\mathcal{C})})$, so that

$$L(E') \text{ is star-free} \iff L(\mathcal{C}) \text{ is star-free} \iff \bigcap_{i \in [n]} L(\mathcal{B}_i) = \emptyset.$$

The simple structure of E' assures that it can be constructed by a logspace-bounded Turing machine. □

Chapter 3

Many-sorted algebra

The study of many-sorted algebra was greatly motivated by the theory of abstract data-types and the algebraic specification of programming languages and their semantics, which arose from the growing need for more efficient (formal) methods in software development. See [23] for a nice survey.

From a universal algebraic point of view, many-sorted (or heterogenous) algebras with *nonempty carrier sets* are natural generalizations of "ordinary" (or homogeneous) algebras, in that many of the concepts and theorems of the classical theory of universal algebra generalize to many-sorted algebras in a straightforward way, as it was pointed out already in the 1960's by Higgins [38] and Birkhoff [10]. For the readers convenience, we summarize here the definitions and theorems of many-sorted algebra which are needed to understand what follows. This summary is based on the papers of Higgins [38], Birkhoff and Lipson [10], Wirsing [51], Manca and Salibra [42] and Guessarian [36].

3.1 Definitions and basic facts

Suppose that S is a set. By an *S-sorted signature* we mean an $S^* \times S$-sorted set Σ of operation symbols. We shall write $\sigma : u \to s$ to indicate that σ is an operation symbol in $\Sigma_{(u,s)}$ $(u \in S^*,\ s \in S)$.

Suppose that Σ is an S-sorted signature. Then an *S-sorted Σ-algebra*, or simply *Σ-algebra*, is a pair $\mathcal{A} = (A, \tau)$, where

A is an S-sorted set (the *universe* of $\mathcal{A}$) such that none of the sets A_s $(s \in S)$ is empty, and

$\tau : \Sigma \to \mathrm{Opn(A)}$ is an $S^* \times S$-sorted function mapping each operation symbol $\sigma \in \Sigma_{(u,s)}$ $(u \in S^*,\ s \in S)$ to a function $\sigma^{\mathcal{A}}_{(u,s)} : A_u \to A_s$.

It is important to stress that we do not allow the carrier sets A_s $(s \in S)$ being empty. We have two good reasons to do so. Firstly, the many-sorted algebras appearing in

this thesis have nonempty carrier sets. Secondly, the generalization of universal algebraic concepts to many-sorted algebras with possibly empty carrier sets is not at all obvious. The best known difficulty was pointed out by Gougen and Meseguer [34]: the "usual" equational deduction rules are unsound with respect to the "usual" concept of validity of equations, where "usual" means straightforward generalization of the corresponding universal algebraic concept to many-sorted algebra. They also proposed a new equational calculus which is sound and complete with respect to the "usual" validity concept. Manca and Salibra [42] approached the problem from the opposite direction. They introduced a new validity concept, called "strong satisfiability", with respect to which the "usual" equational deduction rules form a sound and complete system.

Suppose that X is an S-sorted set of variable symbols such that $X_s \cap \Sigma_{(\epsilon,s)} = \emptyset$, for all $s \in S$.

The collection of Σ-*terms with variables in* X is the least S-sorted set $\mathrm{T}_\Sigma(X)$ such that for all $s \in S$

1. $X_s \subseteq \mathrm{T}_\Sigma(X)_s$, and
2. for all $u \in S^*$, if $\sigma \in \Sigma_{(u,s)}$ and $t \in \mathrm{T}_\Sigma(X)_u$, then $\sigma(t_1, \ldots, t_{|u|}) \in \mathrm{T}_\Sigma(X)_s$. In case $u = \epsilon$ we usually write σ instead of $\sigma()$.

We shall also use *annotated* Σ-*terms with variables in* X. In an annotated term, each variable x is annotated (indexed) by a sort $s \in S$ such that $x \in X_s$, and also each operation symbol σ is indexed by a sort $r \in S$ such that $\sigma \in \Sigma_{(u,r)}$, for some $u \in S^*$.

Formally, the collection of annotated Σ-terms with variables in X is the least S-sorted set $\mathbf{T}_\Sigma(X)$ such that for all $s \in S$

1. if $x \in X_s$ then $x_s \in \mathbf{T}_\Sigma(X)_s$, and
2. for all $u \in S^*$, if $\sigma \in \Sigma_{(u,s)}$ and $t \in \mathbf{T}_\Sigma(X)_u$, then $\sigma_s(t_1, \ldots, t_{|u|}) \in \mathbf{T}_\Sigma(X)_s$. In case $u = \epsilon$ we usually write σ_s instead of $\sigma_s()$.

Note that $\mathbf{T}_\Sigma(X)$ is in fact an S-labeled set, and if X and Σ are S-labeled sets then terms and annotated terms are essentially the same. It turns out that annotated Σ-terms are more important than ordinary ones in the sense that they are the elements of the totally free Σ-algebra. We shall use ordinary Σ-terms only in the so called "meta-equations", see page 43. A single meta-equation is equivalent to a (possibly infinite) collection of annotated equations (i.e., equations between annotated Σ-terms). Thus, some classes of Σ-algebras which are axiomatized by an infinite collection of annotated equations may be axiomatized by a finite number of meta-equations.

The elements of $\mathrm{T}_\Sigma(\emptyset)$ and $\mathbf{T}_\Sigma(\emptyset)$ are called *ground terms*. Following the conventions of [51], we say that Σ is a *sensible signature* if none of the sets $\mathrm{T}_\Sigma(\emptyset)_s$ $(s \in S)$ is empty. Note that in this thesis we only work with sensible signatures.

Suppose now that X is an S-sorted set such that none of the sets $\mathrm{T}_\Sigma(X)_s$ and $\mathbf{T}_\Sigma(X)_s$ ($s \in S$) is empty. (This holds for example when Σ is a sensible signature.) Then we denote by $\mathcal{T}_\Sigma(X)$ (respectively $\boldsymbol{\mathcal{T}}_\Sigma(X)$) the Σ-*algebra of ordinary* (respectively, *annotated*) Σ-*terms* with the operations defined in the natural way:

$$\sigma^{\mathcal{T}_\Sigma(X)}_{(u,s)}(t) = \sigma(t_1, \ldots, t_n),$$
$$\sigma^{\boldsymbol{\mathcal{T}}_\Sigma(X)}_{(u,s)}(r) = \sigma_s(r_1, \ldots, r_n),$$

for all $n \geq 0$, $u \in S^n$, $s \in s$, $t \in \mathrm{T}_\Sigma(X)_u$ and $r \in \mathbf{T}_\Sigma(X)_u$.

Suppose that $\mathcal{A} = (A, \alpha)$ and $\mathcal{B} = (B, \beta)$ are Σ-algebras. An S-sorted relation $\rho \subseteq A \times B$ is called a *simulation from* $\mathcal{A}$ *to* $\mathcal{B}$ if

$$a_1 \,\rho_{u_1}\, b_1 \;\wedge \cdots \wedge\; a_n \,\rho_{u_n}\, b_n \Longrightarrow \sigma^{\mathcal{A}}_{(u,s)}(a) \;\rho_s\; \sigma^{\mathcal{B}}_{(u,s)}(b),$$

for all $n \in \mathbf{N}$, $u \in S^n$, $s \in S$, $\sigma \in \Sigma_{(u,s)}$, $a \in A_u$ and $b \in B_u$. If in addition ρ is an S-sorted function from A to B then we call it a *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$. A surjective homomorphism is called an *epimorphism*, and a bijective homomorphism is called an *isomorphism*. We say that $\mathcal{B}$ is a *homomorphic image* of $\mathcal{A}$ if there exists an epimorphism from $\mathcal{A}$ to $\mathcal{B}$. The algebras $\mathcal{A}$ and $\mathcal{B}$ are called *isomorphic*, denoted $\mathcal{A} \simeq \mathcal{B}$, if there exists an isomorphism from $\mathcal{A}$ to $\mathcal{B}$. A *congruence* on $\mathcal{A}$ is an S-sorted equivalence relation θ on A such that θ is a simulation from $\mathcal{A}$ to $\mathcal{A}$. We denote by $\mathrm{Con}(\mathcal{A})$ the set of all congruences on $\mathcal{A}$. We say that $\mathcal{B}$ is a *subalgebra* of $\mathcal{A}$ if B is an S-sorted subset of A, and the S-sorted identity function $\{\mathrm{id}_{B_s}\}_{s \in S}$ is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. An S-sorted subset of A is called a *subuniverse* of $\mathcal{A}$ if it is the universe of a subalgebra of $\mathcal{A}$. Note that each subalgebra of $\mathcal{A}$ is totally determined by its universe.

Suppose that Σ is a sensible signature, and A_0 is an S-sorted subset of A. Then there exists a smallest subuniverse $\langle A_0 \rangle_{\mathcal{A}}$ of $\mathcal{A}$ containing A_0, called the *subuniverse generated by* A_0 *in* $\mathcal{A}$. The unique subalgebra of $\mathcal{A}$ with universe $\langle A_0 \rangle_{\mathcal{A}}$ is called the *subalgebra generated by* A_0 and denoted also by $\langle A_0 \rangle_{\mathcal{A}}$. We say that A_0 is a *generating system* of $\mathcal{A}$, or equivalently, $\mathcal{A}$ is *generated by* A_0, if $\langle A_0 \rangle_{\mathcal{A}} = \mathcal{A}$.

Suppose that θ is a congruence on $\mathcal{A}$. The *quotient of* $\mathcal{A}$ *under* θ is the Σ-algebra $\mathcal{A}/\theta$ with universe A/θ and operations defined by

$$\sigma^{\mathcal{A}/\theta}_{(u,s)}((a_1\theta_{u_1}) \cdots (a_n\theta_{u_n})) = \sigma^{\mathcal{A}}_{(u,s)}(a)\theta_s$$

for all $n \geq 0$, $u \in S^n$, $s \in S$, $\sigma \in \Sigma_{(u,s)}$ and $a \in A_u$.

Suppose that I is a set of indices, and $\mathcal{A}_i = (A_i, \alpha_i)$ is a Σ-algebra, for each $i \in I$. The *product* of the $\mathcal{A}_i$ is the Σ-algebra

$$\mathcal{P} = \prod_{i \in I} \mathcal{A}_i$$

with universe $P = \prod_{i \in I} A_i$, and operations defined by

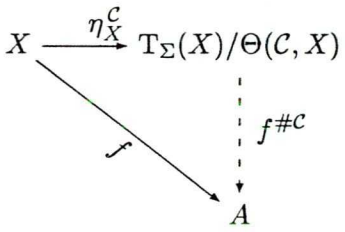
$$(\sigma^{\mathcal{P}}_{(u,s)}(f))(i) = \sigma^{\mathcal{A}_i}_{(u,s)}(f_1(i) \cdots f_n(i))$$

for all $n \geq 0$, $u \in S^n$, $s \in S$, $\sigma \in \Sigma_{(u,s)}$, $i \in I$ and $f \in P_u$.

Suppose that $\mathcal{C}$ is a class of Σ-algebras. Then $\mathbf{H}(\mathcal{C})$, $\mathbf{S}(\mathcal{C})$ and $\mathbf{P}(\mathcal{C})$ respectively denote the classes of all homomorphic images, subalgebras and products of algebras in $\mathcal{C}$. The class $\mathcal{C}$ is called a *variety* of Σ-algebras if it is closed under the operations $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{P}$, i.e., if $\mathbf{H}(\mathcal{C}) \subseteq \mathcal{C}$, $\mathbf{S}(\mathcal{C}) \subseteq \mathcal{C}$ and $\mathbf{P}(\mathcal{C}) \subseteq \mathcal{C}$. The *variety generated by* $\mathcal{C}$ is the smallest variety $\mathbf{V}(\mathcal{C})$ containing $\mathcal{C}$. Just as for ordinary algebras, we have

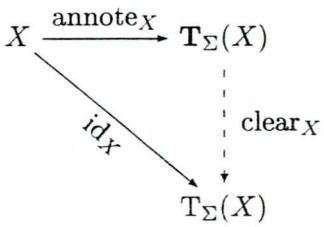
$$\mathbf{V}(\mathcal{C}) = \mathbf{HSP}(\mathcal{C}).$$

Suppose that X is an S-sorted set, Σ is a sensible S-sorted signature, and $\mathcal{C}$ is a class of Σ-algebras. We denote by $\mathcal{F}_{\mathcal{C}}(X)$ the Σ-algebra $\mathrm{T}_\Sigma(X)/\Theta(\mathcal{C}, X)$, where $\Theta(\mathcal{C}, X)$ is the intersection of the kernels of all homomorphisms $h : \mathrm{T}_\Sigma(X) \to \mathcal{A}$, for all algebras $\mathcal{A} \in \mathcal{C}$. Then $\mathcal{F}_{\mathcal{C}}(X)$ has the property that for each algebra $\mathcal{A} = (A, \alpha)$ in $\mathcal{C}$, and for each S-sorted function $f : X \to A$, there exists a unique homomorphism $f^{\#\mathcal{C}} : \mathcal{F}_{\mathcal{C}}(X) \to \mathcal{A}$ such that the diagram



commutes (that is, $f_s^{\#\mathcal{C}}(\eta_{X,s}^{\mathcal{C}}(x)) = f_s(x)$, for all $s \in S$ and $x \in X_s$), where $\eta_X^{\mathcal{C}}$ is the S-sorted function mapping each element $x \in X_s$ to the congruence class $x\,\Theta(\mathcal{C}, X)_s$. When $\mathcal{C}$ is a variety the algebra $\mathcal{F}_{\mathcal{C}}(X)$ itself belongs to $\mathcal{C}$. In this case we call $\mathcal{F}_{\mathcal{C}}(X)$ *the X-generated free algebra in $\mathcal{C}$*. Since we identify isomorphic algebras, any algebra isomorphic to $\mathcal{F}_{\mathcal{C}}(X)$ is also called the X-generated free algebra in $\mathcal{C}$. In particular, if $\mathcal{C}$ is the class of all Σ-algebras then the X-generated free algebra in $\mathcal{C}$ is isomorphic to the term algebra $\mathrm{T}_\Sigma(X)$. In this case we write $f^{\#}$ for $f^{\#\mathcal{C}}$.

Let $\mathrm{annote}_X : X \to \mathbf{T}_\Sigma(X)$ denote the S-sorted function mapping each variable $x \in X_s$ to the annotated Σ-term x_s. Then there exists a unique homomorphism $\mathrm{clear}_X : \mathcal{T}_\Sigma(X) \to \mathcal{T}_\Sigma(X)$ such that the diagram



is commutative. When X is understood we simply write annote for annote_X and clear for clear_X.

Lemma 3.1.1. *Suppose that $\mathcal{C}$ is a class of Σ-algebras, X is an S-sorted set. $\mathcal{A}, \mathcal{B} \in \mathcal{C}$, $f : X \to A$ is an S-sorted function, and $g : \mathcal{A} \to \mathcal{B}$ is a homomorphism. Then $f^{\#\mathcal{C}} \circ g = (f \circ g)^{\#\mathcal{C}}$.*

Proof. We have $\eta^{\mathcal{C}}_X \circ f^{\#\mathcal{C}} \circ g = f \circ g$. Since $(f \circ g)^{\#\mathcal{C}}$ is the only homomorphism $h : \mathcal{F}_{\mathcal{C}}(X) \to \mathcal{A}$ with $\eta^{\mathcal{C}}_X \circ h = f \circ g$ it follows that $f^{\#\mathcal{C}} \circ g = (f \circ g)^{\#\mathcal{C}}$. □

Suppose that X is an S-sorted set of variables. By an *annotated equation of sort* $s \in S$ *with variables in* X we mean an ordered pair (t_1, t_2) of annotated Σ-terms $t_1, t_2 \in \mathbf{T}_\Sigma(X)_s$, usually written in the form $t_1 = t_2$. We say that *the annotated equation* $t_1 = t_2$ *is valid in the algebra* $\mathcal{A} = (A, \alpha)$, or equivalently, the algebra $\mathcal{A}$ *satisfies the equation* $t_1 = t_2$, or $\mathcal{A}$ is a *model of the equation* $t_1 = t_2$, denoted $\mathcal{A} \models t_1 = t_2$, if $f^{\#}_s(t_1) = f^{\#}_s(t_2)$ holds for all S-sorted functions $f : X \to A$.

Suppose that $\mathcal{C}$ is a class of S-sorted Σ-algebras and E is an S-sorted set of annotated equations. We write $\mathcal{C} \models E$ if each algebra in $\mathcal{C}$ satisfies each equation in E. We denote by $\mathbf{Eq}_X(\mathcal{C})$ the S-sorted set of those annotated equations with variables in X which are valid in each algebra in $\mathcal{C}$. The *equational theory* of $\mathcal{C}$ is the S-sorted set $\mathbf{Eq}(\mathcal{C}) := \mathbf{Eq}_{\mathbf{X}}(\mathcal{C})$, where $\mathbf{X}$ is a fixed S-sorted set of variables such that each $\mathbf{X}_s$ ($s \in S$) is countably infinite. We let $\mathbf{Mod}(E)$ denote the class of those Σ-algebras which satisfy all equations in E. An annotated equation e is called a *logical consequence* of E, denoted $E \models e$, if $\mathbf{Mod}(E) \models e$. Note that $\mathbf{Eq}_X(\mathbf{Mod}(E))$ is the set of all logical consequences of E with variables in X. We say that E *(equationally) axiomatizes* the class $\mathcal{C}$ if $\mathcal{C} = \mathbf{Mod}(E)$. Conversely, $\mathcal{C}$ is called an *equational class* of Σ-algebras if it is axiomatized by a set of annotated equations. Note that $\mathcal{C}$ is equational if and only if $\mathcal{C} = \mathbf{Mod}(\mathbf{Eq}_X(\mathcal{C}))$ for some S-sorted set X, or equivalently, if $\mathcal{C} = \mathbf{Mod}(\mathbf{Eq}(\mathcal{C}))$. It was proved in [38] that Birkhoff's theorem generalizes to many-sorted algebras: a class of Σ-algebras is equational if and only if it is a variety.

Suppose that $t \in \mathbf{T}_\Sigma(X)_s$ and $t' \in \mathbf{T}_\Sigma(X)_r$ are annotated Σ-terms, and $x \in X_r$ is a variable. Then $t[t'/x_r]$ denotes the annotated Σ-term which we get by substituting t' for each occurrence of the annotated variable x_r in t. The *deductive closure* of a set E of annotated equations with variables in X is the least S-sorted set $\mathbf{D}(E)$ of annotated equations such that for all $s, r \in S$ and $x \in X_r$

$$
\begin{aligned}
e \in E_s &\Longrightarrow e \in \mathbf{D}(E)_s \\
t \in \mathbf{T}_\Sigma(X)_s &\Longrightarrow t = t \in \mathbf{D}(E)_s \\
t_1 = t_2 \in \mathbf{D}(E)_s &\Longrightarrow t_2 = t_1 \in \mathbf{D}(E)_s \\
t_1 = t_2 \in \mathbf{D}(E)_s \ \wedge\ t_2 = t_3 \in \mathbf{D}(E)_s &\Longrightarrow t_1 = t_3 \in \mathbf{D}(E)_s \\
t_1 = t_2 \in \mathbf{D}(E)_s \ \wedge\ t_3 = t_4 \in \mathbf{D}(E)_r &\Longrightarrow t_1[t_3/x_r] = t_2[t_4/x_r] \in \mathbf{D}(E)_s
\end{aligned}
$$

An annotated equation e is called a *syntactical consequence* of a set E of annotated equations, denoted $E \vdash e$, if $e \in \mathbf{D}(E)$. It follows from Theorem 4.5 of [42] that the above notions of logical and syntactical consequence of annotated equations are the same, i.e., $E \models e \iff E \vdash e$.

We shall also use *meta-equations* of sort $s \in S$, which are equations of the form $t_1 \approx t_2$, where $t_1, t_2 \in \mathrm{T}_\Sigma(X)_s$ are ordinary Σ-terms of sort s. With each meta-equation $t_1 \approx t_2$ we associate an S-sorted set $\mathbf{Eq}(t_1 \approx t_2)$ of annotated equations defined as follows. For all sorts $r \in S$, and for all annotated Σ-terms $t'_1, t'_2 \in \mathbf{T}_\Sigma(X)_r$, the annotated equation $t'_1 = t'_2$ belongs to $\mathbf{Eq}(t_1 \approx t_2)_r$ if and only if $\mathrm{clear}_r(t'_1) = t_1$,

$\mathrm{clear}_r(t_2') = t_2$, and if x_a and x_b $(a, b \in S)$ are two annotated occurrences of the same variable symbol $x \in X_a \cap X_b$ in t_1' or t_2' then $a = b$.

We say that a *meta-equation* $t_1 \approx t_2$ *is valid in* a Σ-*algebra* $\mathcal{A} = (A, \alpha)$, or equivalently, $\mathcal{A}$ *satisfies the meta-equation* $t_1 \approx t_2$, or $\mathcal{A}$ is a *model of the meta-equation* $t_1 \approx t_2$, denoted $\mathcal{A} \models t_1 \approx t_2$, if $\mathcal{A} \models \mathbf{Eq}(t_1 \approx t_2)$.

Suppose that E is an S-sorted set of meta-equations. Then we define $\mathbf{Eq}(E) := \bigcup_{e \in E} \mathbf{Eq}(e)$, and $\mathrm{Mod}(E) := \mathrm{Mod}(\mathbf{Eq}(E))$. When $\mathcal{C}$ is a class of Σ-algebras we write $\mathcal{C} \models E$ if $\mathcal{C} \models \mathbf{Eq}(E)$.

3.2 From categories to iteration theories

This section is devoted to preiteration theories, some enriched $\mathbf{N}$-categories with coproducts, first defined by Bloom, Elgot, and Wright in [11] and [12], and Ésik [15]. We show two possible ways of representing preiteration theories as varieties of $\mathbf{N} \times \mathbf{N}$-sorted algebras. The two representations use different signatures. According to the first representation (which is based on an infinitary signature), the class of preiteration theories is axiomatized by an infinite number of meta-equations. The second representation of preiteration theories is based on a finitary signature, which also makes it possible to give a finite meta-equational axiomatization.

3.2.1 Categories

A (small) category $\mathcal{C}$ consists of a set of *objects*, and for each pair a, b of objects, a set of *morphisms* with source a and target b. We write $f : a \to b$ to indicate that f is a morphism with source a and target b. A category is equipped with a binary operation of composition mapping each pair $f : a \to b$, $g : b \to c$ of morphisms to a morphism $f \cdot g : a \to c$, for all objects a, b, c. A category also has *identity morphisms* $\mathbf{1}_a : a \to a$, for all objects a. The composition operation is required to be associative, when defined, and the morphisms $\mathbf{1}_a$ are neutral elements with respect to composition. Thus the equations

$$\begin{aligned} (f \cdot g) \cdot h &= f \cdot (g \cdot h) \\ \mathbf{1}_a \cdot f &= f \\ f \cdot \mathbf{1}_b &= f \end{aligned}$$

hold in any category $\mathcal{C}$, for all objects a, b, c, d, and morphisms $f : a \to b$, $g : b \to c$, and $h : c \to d$.

An $\mathbf{N}$-*category* is a category whose objects are the nonnegative integers. An *algebraic theory*, or *theory* for short, is an $\mathbf{N}$-category T such that for each $n \in \mathbf{N}$, there are n distinguished morphisms $1_n, 2_n, \ldots, n_n$ with source 1 and target n, called *coproduct injections*, with the following *coproduct property*. For any object $n, p \in \mathbf{N}$, and any (possibly empty) family $f_1, \ldots, f_n$ of morphisms $1 \to p$, there is a unique morphism $f : n \to p$ such that $i_n \cdot f = f_i$, for all $i \in [n]$. The morphism f determined by

the family $f_1, \ldots, f_n$ is called the *(source) tupling* of the family, and is denoted $\langle f_1, \ldots, f_n \rangle$.

Thus, in any theory T, the coproduct property determines a *tupling* operation on the morphisms. This operation can be applied to any family $f_1, \ldots, f_n$ of morphisms $1 \to p$, and it yields a morphism $\langle f_1, \ldots, f_n \rangle$ with source n and target p. In case $n = 0$, the empty family of morphisms $1 \to p$ determines a unique morphism with source 0 and target p, which we denote by 0_p, for all $p \in \mathbf{N}$.

Thus, if T is a theory then the equations

$$i_n \cdot \langle f_1, \ldots, f_n \rangle = f_i \tag{3.1}$$

$$\langle 1_n \cdot f, \ldots, n_n \cdot f \rangle = f \tag{3.2}$$

hold in T, for all objects $n, p \in \mathbf{N}$, integers $i \in [n]$, morphisms $f : n \to p$ and $g : 0 \to p$, and any family $f_1, \ldots, f_n$ of morphisms $1 \to p$.

It is also required that the coproduct injection $1_1 : 1 \to 1$ be the same as the identity morphism $\mathbf{1}_1$, so that any theory T satisfies the equation

$$\mathbf{1}_1 = 1_1 \tag{3.3}$$

by definition.

Now suppose that T is an $\mathbf{N}$-category equipped with a tupling operation and distinguished morphisms $i_n : 1 \to n$. Suppose moreover that T satisfies the equations (3.1–3.3). Then it follows that T is a theory. Indeed, since we assumed that T satisfies (3.3) we only need to show that T has the coproduct property. Let the coproduct injections be the morphisms i_n ($n > 0$, $i \in [n]$). Then, given any (possibly empty) family $f_1, \ldots, f_n$ of morphisms $1 \to p$ in T, equation (3.1) assures that there exists a morphism $f : n \to p$ in T such that $i_n \cdot f = f_i$, for all $i \in [n]$. Indeed, the tupling operation applied to $f_1, \ldots, f_n$ yields such a morphism. On the other hand, equation (3.2) asserts that $\langle f_1, \ldots, f_n \rangle$ is the only such morphism.

Thus we may say that a theory is an $\mathbf{N}$-category equipped with a tupling operation and distinguished morphisms i_n ($n > 0$, $i \in [n]$) satisfying the equations (3.1–3.3).

It follows that the equations

$$\mathbf{1}_n = \langle 1_n, \ldots, n_n \rangle \tag{3.4}$$

$$f = \langle f \rangle \tag{3.5}$$

hold in any theory T, for all objects $n, p \in \mathbf{N}$ and morphisms $f : 1 \to p$.

A morphism is called a *base morphism* if it is the source tupling of a (possibly empty) family of coproduct injections. Note that the identity morphisms are base morphisms by equation (3.4), and also each coproduct injection itself is a base morphism by equation (3.5). A theory T is called *nontrivial* if the two base morphisms 1_2 and 2_2 are different in T. It is easy to see that T is a trivial theory if and only if it has at most one morphism $n \to p$, for all $n, p \in \mathbf{N}$. In fact, if T is a trivial theory, and $p > 0$ or $n = 0$, then T has a unique morphism $n \to p$. A theory has a morphism $n \to 0$ for each $n > 0$ if and only if it has at least one morphism $1 \to 0$.

A subcategory T_0 of a theory T is called a *subtheory of* T if T_0 is a theory with the same coproduct injections as T. It is not hard to see that the collection of base morphisms is closed under the composition operation, so that base morphisms form a subtheory in any theory T. Moreover, if T is a nontrivial theory then the subtheory of base morphisms in T is isomorphic to the theory **Tot** of all (total) functions $[n] \to [p]$. Composition in **Tot** is function composition, and the base morphism $\langle i_p^{(1)}, \ldots, i_p^{(n)} \rangle$ $(n, p \in \mathbf{N},\ i^{(1)}, \ldots, i^{(n)} \in [p])$ is the function mapping each element $j \in [n]$ to $i^{(j)} \in [p]$. We call a base morphism $\rho : n \to p$ *surjective* (respectively, *injective*) if the corresponding function $\rho : [n] \to [p]$ is surjective (injective, respectively).

The tupling operation can be extended to morphisms having a common target but arbitrary source as follows. For any objects $n, p, k_1, \ldots, k_n \in \mathbf{N}$, and for any family of morphisms $f^{(i)} : k_i \to p$, $i \in [n]$, we define

$$\langle f^{(1)}, \ldots, f^{(n)} \rangle := \langle f_1^{(1)}, \ldots, f_{k_1}^{(1)}, \ldots, f_1^{(n)}, \ldots, f_{k_n}^{(n)} \rangle, \tag{3.6}$$

where $f_j^{(i)}$ is an abbreviation for $j_{k_i} \cdot f^{(i)}$. In the special case $n = 2$ this operation is called *pairing*.

Using the generalized tupling operation (as well as composition and the coproduct injections) we may define another binary operation called *separated sum* as follows. For all objects $n, p, m, q \in \mathbf{N}$, and all morphisms $f : n \to p$ and $g : m \to q$, let the separated sum of f and g be the morphism

$$f \oplus g = \langle f \cdot \kappa_{p,q},\ g \cdot \lambda_{p,q} \rangle \tag{3.7}$$

where $\kappa_{p,q} : p \to p + q$ and $\lambda_{p,q} : q \to p + q$ are base morphisms defined by

$$\kappa_{p,q} = \langle 1_{p+q}, \ldots, p_{p+q} \rangle \tag{3.8}$$

$$\lambda_{p,q} = \langle (p+1)_{p+q}, \ldots, (p+q)_{p+q} \rangle. \tag{3.9}$$

Note that $f \oplus g$ has source $n + m$ and target $p + q$.

A *preiteration theory* is a theory equipped with an additional operation called *iteration* or *dagger*, mapping each morphism $f : n \to n + p$ to a morphism $f^\dagger : n \to p$. Note that while a theory may not have any morphisms with target 0, a preiteration theory always contains the morphisms $\mathbf{1}_n^\dagger : n \to 0$, for all $n \in \mathbf{N}$.

3.2.2 Preiteration theories as many-sorted algebras

Let PT be the $(\mathbf{N} \times \mathbf{N})$-sorted signature consisting of the following operation symbols.

$$\begin{array}{rrcl} \texttt{comp}: & (n,p)\,(p,q) & \to & (n,q) \\ \texttt{tupl}: & (1,p)^n & \to & (n,p) \\ \texttt{iter}: & (n, n+p) & \to & (n,p) \\ \mathbf{1}: & \epsilon & \to & (n,n) \\ i: & \epsilon & \to & (1,r) \end{array}$$

for all $n, m, p, q, \in \mathbf{N}$, $r > 0$, and $i \in [r]$.

Clearly, if T is an $\mathbf{N}$-category equipped with the additional operations of tupling and iteration of morphisms, as well as distinguished morphism $i_n : 1 \to n$ ($n > 0$, $i \in [n]$), then T may be considered as an $\mathbf{N} \times \mathbf{N}$-sorted PT-algebra: the elements of sort (n,p) are all morphism with source n and target p, and the operation symbols `comp`, `tupl` and `iter` are interpreted as the composition, tupling and iteration operations, respectively. The operation symbol $\mathbf{1} \in \mathrm{PT}_{(n,n)}$ is interpreted as the identity morphism $\mathbf{1} : n \to n$, and $i \in \mathrm{PT}_{(1,n)}$ is interpreted as the coproduct injection $i_n : 1 \to n$.

We shall usually omit the annotations of the operation symbols `comp` and `iter` in PT-terms, because the sort of the result produced by these operations is uniquely determined by the sorts of the operands. For example, if t is an annotated PT-term of sort (n,p) and t' is an annotated PT-term of sort (p,q) then $t \cdot t'$ is of sort (n,q). For the same reason we omit the annotation of the operation symbol `tupl` when it is applied to a nonempty family of annotated terms. (Note that the annotation may not be omitted when `tupl` is applied to the empty family of annotated terms of sort $(0,p)$, since in this case we must indicate that the resulting term is of sort $(0,p)$.) Moreover, we abbreviate the annotated PT-term $\mathbf{1}_{(n,n)}$ as $\mathbf{1}_n$, $i_{(1,n)}$ as i_n ($n > 0$, $i \in [n]$), $\mathtt{comp}(f,g)$ as $f \cdot g$, $\mathtt{tupl}(t_1, \ldots, t_n)$ as $\langle t_1, \ldots, t_n \rangle$, and $\mathtt{iter}(f)$ as $f^\dagger$.

It follows by definition that the PT-algebra T satisfies the annotated equations

$$\begin{aligned} (f_{(n,m)} \cdot g_{(m,p)}) \cdot h_{(p,q)} &= f_{(n,m)} \cdot (g_{(m,p)} \cdot h_{(p,q)}) \\ \mathbf{1}_n \cdot f_{(n,p)} &= f_{(n,p)} \\ f_{(n,p)} \cdot \mathbf{1}_p &= f_{(n,p)} \end{aligned}$$

for all $n, m, p, q \in \mathbf{N}$, where f, g, h are variables of sort (n,p) for all $n, p \in \mathbf{N}$. *From now on the variables appearing in meta-equations or annotated equations are assumed to have arbitrary sorts, unless otherwise stated.*

The reader may verify easily that this infinite collection of annotated equations is equivalent to the following three (familiarly looking) meta-equations.

$$(f \cdot g) \cdot h \approx f \cdot (g \cdot h) \tag{3.10}$$

$$\mathbf{1} \cdot f \approx f \tag{3.11}$$

$$f \cdot \mathbf{1} \approx f. \tag{3.12}$$

We shall call these meta-equations the *category identities*, since they express the fact that the PT-algebra T is a category. It follows by our discussion in the previous subsection that T is a preiteration theory if and only if T is a PT-algebra satisfying the category identities and the following (infinite) collection of meta-equations.

$$i \cdot \langle f^{(1)}, \ldots, f^{(n)} \rangle \approx f^{(i)} \tag{3.13}$$

$$\langle 1 \cdot f, \ldots, n \cdot f \rangle \approx f \tag{3.14}$$

$$1 \approx 1 \tag{3.15}$$

for all $n \in \mathbf{N}$ and $i \in [n]$.

Thus, preiteration theories form a variety of PT-algebras axiomatized by an infinite collection of meta-equations. However, it would be nicer to have a finite axiomatization. Another flaw of this representation of preiteration theories is that the signature PT is not finitary. One may ask if we can get rid of these unpleasant properties by choosing some alternative representation. The answer is yes.

Let Θ be the $(\mathbf{N} \times \mathbf{N})$-sorted signature consisting of the operation symbols

$$\begin{array}{rrcl}
\mathtt{pair}: & (n,p)\,(m,p) & \to & (n+m,p) \\
\mathtt{comp}: & (n,p)\,(p,q) & \to & (n,q) \\
\mathtt{iter}: & (n,n+p) & \to & (n,p) \\
\kappa: & \epsilon & \to & (p,p+q) \\
\lambda: & \epsilon & \to & (q,p+q) \\
\mathbf{1}: & \epsilon & \to & (n,n) \\
0: & \epsilon & \to & (0,n)
\end{array}$$

for all $n, m, p, q, \in \mathbf{N}$. Note that Θ (just as PT) is a sensible signature, and it is also finitary. The new operation symbols $\mathtt{pair}$, κ, λ, and 0 correspond to the pairing operation, and the distinguished morphisms $\kappa_{p,q}$, $\lambda_{p,q}$, and 0_p of preiteration theories. For these operations and constants we adopt similar notational conventions as for the others. In particular, we omit unnecessary indices and write $\langle f, g\rangle$ for $\mathtt{pair}(f,g)$, $\kappa_{p,q}$ for $\kappa_{(p,p+q)}$, $\lambda_{p,q}$ for $\lambda_{(q,p+q)}$, and 0_p for $0_{(0,p)}$.

Any PT-algebra $\mathcal{T}$ can be transformed into a Θ-algebra $\Theta(\mathcal{T})$ by defining the pairing operation by the special case $n = 2$ of equation (3.6), the constant κ by equation (3.8), λ by equation (3.9), and 0_p as $\langle\,\rangle_p$, for all $p \in \mathbf{N}$. Obviously, the interpretation of the composition and iteration operations, and the constant $\mathbf{1}$ remain the same as in $\mathcal{T}$.

We also define the converse transformation as follows. Given an arbitrary Θ-algebra $\mathcal{T}$, let $\mathrm{PT}(\mathcal{T})$ be the PT-algebra in which the tupling operation and the constants i are defined by

$$\langle\,\rangle_p = 0_p \tag{3.16}$$

$$\langle f_1\rangle = f_1 \tag{3.17}$$

$$\langle f_1, f_2, \ldots, f_m\rangle = \langle f_1, \langle f_2, \ldots \langle f_{m-1}, f_m\rangle \ldots\rangle\rangle \tag{3.18}$$

$$i_n = \kappa_{1,n-i} \cdot \lambda_{i-1,n-i+1}, \tag{3.19}$$

for all $p \in \mathbf{N}$, $m \geq 2$, $n \geq 1$, and $i \in [n]$.

The following was proved by Stephen L. Bloom in an unpublished manuscript.

Theorem 3.2.1. *If $\mathcal{T}$ is a* PT*-algebra satisfying the category identities* (3.10–3.12) *and the meta-equations* (3.13–3.15)*, then $\Theta(\mathcal{T})$ satisfies the following meta-equations, called the preiteration theory identities.*

$$f \cdot (g \cdot h) \approx (f \cdot g) \cdot h \tag{3.20}$$

$$\mathbf{1} \cdot f \approx f \tag{3.21}$$

$$f \cdot \mathbf{1} \approx f \tag{3.22}$$

$$\langle f, \langle g, h\rangle\rangle \approx \langle\langle f, g\rangle, h\rangle \tag{3.23}$$
$$\langle f, 0\rangle \approx f \tag{3.24}$$
$$\langle 0, f\rangle \approx f \tag{3.25}$$
$$\kappa \cdot \kappa \approx \kappa \tag{3.26}$$
$$\lambda \cdot \lambda \approx \lambda \tag{3.27}$$
$$\langle \kappa, \lambda\rangle \approx \mathbf{1} \tag{3.28}$$
$$\kappa \approx \mathbf{1} \tag{3.29}$$
$$\lambda \approx \mathbf{1} \tag{3.30}$$
$$\kappa \cdot \langle f, g\rangle \approx f \tag{3.31}$$
$$\lambda \cdot \langle f, g\rangle \approx g \tag{3.32}$$
$$\langle \kappa \cdot f, \lambda \cdot f\rangle \approx f \tag{3.33}$$
$$0 \approx f \tag{3.34}$$

Conversely, if $\mathcal{T}$ is a Θ-algebra satisfying the preiteration theory identities then the PT*-algebra* $\mathrm{PT}(\mathcal{T})$ *satisfies the category identities and the meta-equations* (3.13–3.15). □

For the reader's convenience we spell the annotated versions of the preiteration theory identities.

$$f_{(n,p)} \cdot (g_{(p,q)} \cdot h_{(q,r)}) = (f_{(n,p)} \cdot g_{(p,q)}) \cdot h_{(q,r)} \tag{3.20′}$$
$$\mathbf{1}_n \cdot f_{(n,p)} = f_{(n,p)} \tag{3.21′}$$
$$f_{(n,p)} \cdot \mathbf{1}_p = f_{(n,p)} \tag{3.22′}$$
$$\langle f_{(n,p)}, \langle g_{(m,p)}, h_{(s,p)}\rangle\rangle = \langle\langle f_{(n,p)}, g_{(m,p)}\rangle, h_{(s,p)}\rangle \tag{3.23′}$$
$$\langle f_{(n,p)}, 0_p\rangle = f_{(n,p)} \tag{3.24′}$$
$$\langle 0_p, f_{(n,p)}\rangle = f_{(n,p)} \tag{3.25′}$$
$$\kappa_{p,q} \cdot \kappa_{p+q,r} = \kappa_{p,q+r} \tag{3.26′}$$
$$\lambda_{p,q} \cdot \lambda_{r,q+p} = \lambda_{p+r,q} \tag{3.27′}$$
$$\langle \kappa_{p,q}, \lambda_{p,q}\rangle = \mathbf{1}_{p+q} \tag{3.28′}$$
$$\kappa_{p,0} = \mathbf{1}_p \tag{3.29′}$$
$$\lambda_{0,q} = \mathbf{1}_q \tag{3.30′}$$
$$\kappa_{n,m} \cdot \langle f_{(n,p)}, g_{(m,p)}\rangle = f_{(n,p)} \tag{3.31′}$$
$$\lambda_{n,m} \cdot \langle f_{(n,p)}, g_{(m,p)}\rangle = g_{(m,p)} \tag{3.32′}$$
$$\langle \kappa_{n,m} \cdot f_{(n+m,p)}, \lambda_{n,m} \cdot f_{(n+m,p)}\rangle = f_{(n+m,p)} \tag{3.33′}$$
$$0_p = f_{(0,p)} \tag{3.34′}$$

for all $n, m, p, q, r, s \in \mathbf{N}$.

From now on by a *preiteration theory* we mean a Θ-algebra satisfying the preiteration theory identities, so that the class of all preiteration theories is a variety of Θ-algebras, which we denote by $\mathrm{TH}^{\dagger}$.

Thus, the „legal" operations in a preiteration theory are pairing, composition, iteration, and the constants κ, λ, $\mathbf{1}$ and 0. For convenience however we shall continue using the tupling operation and the constants i_n, keeping in mind that they are abbreviations defined by (3.16–3.19). We shall also use the *separated sum* operation defined by (3.7). We consider $\oplus$ as an infix binary operation symbol of sort $(n,p)(m,q) \to (n+m, p+q)$, for all $n,m,p,q \in \mathbf{N}$. The precedence of $\oplus$ is lower than the precedence of $\cdot$, so that $f \cdot g \oplus h$ means $(f \cdot g) \oplus h$.

It is well known (see [15]) that the following meta-equations hold in any preiteration theory:

$$f \oplus (g \oplus h) \approx (f \oplus g) \oplus h \tag{3.35}$$
$$0 \oplus f \approx f \tag{3.36}$$
$$f \oplus 0 \approx f \tag{3.37}$$
$$\mathbf{1} \oplus 0 \approx \kappa \tag{3.38}$$
$$0 \oplus \mathbf{1} \approx \lambda \tag{3.39}$$
$$\langle f, g \rangle \cdot h \approx \langle f \cdot h, g \cdot h \rangle \tag{3.40}$$
$$(f \oplus g) \cdot \langle h_1, h_2 \rangle \approx \langle f \cdot h_1, g \cdot h_2 \rangle \tag{3.41}$$
$$(f \oplus g) \cdot (h_1 \oplus h_2) \approx (f \cdot h_1) \oplus (g \cdot h_2) \tag{3.42}$$
$$0 \cdot f \approx 0 \tag{3.43}$$
$$f \cdot \kappa \approx f \oplus 0 \tag{3.44}$$
$$f \cdot \lambda \approx 0 \oplus f \tag{3.45}$$

where f, g, h, h_1, h_2 are variables of arbitrary sorts.

3.2.3 The preiteration theory of partial functions

We have already seen an example of a theory: **Tot**, the theory of all functions $[n] \to [p]$. If we want to turn **Tot** into a preiteration theory we have to define an iteration operation on its elements. However, this is impossible, since **Tot** has no element of sort $(1,0)$ (there is no total function from $[1]$ to $\emptyset$), and thus we cannot define $\mathbf{1}_1^\dagger$. A possible solution to this problem is to introduce all partial functions $[n] \to [p]$ as elements of sort (n,p).

Let **Pfn** (for partial functions) denote the Θ-algebra in which the elements of sort (n,p) are all partial function from $[n]$ to $[p]$. The constants $\mathbf{1}_n$, $\kappa_{p,q}$, $\lambda_{p,q}$ and 0_n in **Pfn** are the same as in **Tot**. In particular, $\mathbf{1}_n$ is the identity function $\mathrm{id}_{[n]} : [n] \to [n]$, 0_n is the empty function $\emptyset : [0] \to [n]$, $\kappa_{p,q}$ is the identity function $\mathrm{id}_{[p]} : [p] \to [p+q]$, and $\lambda_{p,q} : [q] \to [p+q]$ is the function mapping each element i in $[q]$ to $p+i \in [p+q]$. Composition in **Pfn** is composition of partial functions. The pairing operation is defined by

$$i \langle f, g \rangle = \begin{cases} i f & \text{if } i \leq n \\ (i-n) g & \text{otherwise.} \end{cases}$$

for all partial functions $f : [n] \to [p]$, $g : [m] \to [p]$, and integer $i \in [n+m]$. The iteration operation applied to a partial function $f : [n] \to [n+p]$ yields the partial function $f^\dagger : [n] \to [p]$ defined by

$$i\, f^\dagger\, j \iff i\, f^+\, n + j,$$

for all $i \in [n]$ and $j \in [p]$, where $f^+ : [n] \to [n+p]$ is the transitive closure of f. Thus, $(\mathbf{1}_n)^\dagger$ is the empty partial function from $[n]$ to $[0]$, for all $n \geq 0$.

3.2.4 Conway theories

A *Conway theory* [15] is a preiteration theory satisfying the following meta-equations called *Conway identities*:

parameter identity

$$(f \cdot (\mathbf{1} \oplus g))^\dagger \approx f^\dagger \cdot g, \tag{3.46}$$

double dagger identity

$$(f \cdot (\langle \mathbf{1}, \mathbf{1} \rangle \oplus \mathbf{1}))^\dagger \approx f^{\dagger\dagger}, \tag{3.47}$$

composition identity

$$(f \cdot \langle g, \lambda \rangle)^\dagger \approx f \cdot \langle (g \cdot \langle f, \lambda \rangle)^\dagger,\ \mathbf{1} \rangle, \tag{3.48}$$

The term "Conway identities" comes from the form these identities take in matrix theories over semirings equipped with a * operation, see [21] or the books [15, 35]. For example, the double dagger identity corresponds to the equation

$$(a+b)^* = (a^* b)^* a^*$$

and the composition identity to the equation

$$(ab)^* = a(ba)^* b + 1.$$

Note that every Conway theory satisfies Elgot's *fixed point identity* [25]

$$f^\dagger \approx f \cdot \langle f^\dagger, \mathbf{1} \rangle. \tag{3.49}$$

In *-semirings the fixed point identity takes the form

$$a^* = aa^* + 1.$$

Thus, Conway theories form a variety of Θ-algebras. It is well known that the preiteration theory **Pfn** is freely generated by the empty $\mathbf{N} \times \mathbf{N}$-sorted set in the variety of all Conway theories, see [15].

3.2.5 Iteration theories

In the sequel we shall call the elements of the initial Θ-algebra $\mathcal{T}_\Theta(\emptyset)$ *partial base terms*. We call them "base" because they form the smallest term algebra over the signature Θ, and we call them "partial" because any such term $t : n \to p$ denotes a partial function from $[n]$ to $[p]$, namely the partial function $\iota^{\mathbf{Pfn}}_{(n,p)}(t)$, where $\iota^{\mathbf{Pfn}}$ is the unique homomorphism from $\mathcal{T}_\Theta(\emptyset)$ to the Θ-algebra $\mathbf{Pfn}$. A partial base term $t : n \to p$ is called a *base term* if $\iota^{\mathbf{Pfn}}_{(n,p)}(t)$ is a total function from $[n]$ to $[p]$.

A base term $t : n \to p$ is called *injective* (respectively, *surjective*) if $\iota^{\mathbf{Pfn}}_{(n,p)}(t)$ is an injective (surjective) function.

A Conway theory $\mathcal{T}$ is called an *iteration theory* if it satisfies the following collection of annotated Θ-equations, the *commutative identities* [29]:

$$\langle 1_m \cdot \rho \cdot f_{(n,m+p)} \cdot (\rho_1 \oplus \mathbf{1}_p), \ldots, m_m \cdot \rho \cdot f_{(n,m+p)} \cdot (\rho_m \oplus \mathbf{1}_p)\rangle^\dagger = \rho \cdot (f_{(n,m+p)} \cdot (\rho \oplus \mathbf{1}_p))^\dagger,$$

where f is a variable, ρ is a surjective base term of sort (m, n), and $\rho_1, \ldots, \rho_m$ are base terms of sort (m, m) such that the equations $\rho_i \cdot \rho = \rho$ hold in $\mathbf{Pfn}$, for all $i \in [m]$.

Recently Zoltán Ésik [30] has given another axiomatization of iteration theories. The new axiomatization consists of the Conway identities and an annotated Θ-equation associated with each (simple) finite group. Suppose that $\mathcal{G} = ([n]; 1, \circ)$ is a (simple) group on the first n positive integers with identity element 1. Then the *group identity* associated with $\mathcal{G}$ is the annotated Θ-equation

$$(f_{(1,n)} \cdot (\tau \oplus \mathbf{1}_p))^\dagger = 1_n \cdot \langle f_{(1,n)} \cdot (\rho^{\mathcal{G}}_1 \oplus \mathbf{1}_p), \ldots, f_{(1,n)} \cdot (\rho^{\mathcal{G}}_n \oplus \mathbf{1}_p)\rangle^\dagger,$$

where f is a variable, τ is the annotated base term $\langle 1_n, \ldots, 1_n\rangle$ of sort (n, n), and $\rho^{\mathcal{G}}_i$ is the annotated base term $\langle (i \circ 1)_n, (i \circ 2)_n, \ldots, (i \circ n)_n\rangle$ of sort (n, n), for each $i \in [n]$.

3.2.6 Signatures vs. $\mathbf{N} \times \mathbf{N}$-sorted sets

Any signature (that is, $\mathbf{N}$-labeled set) may be considered as an $\mathbf{N} \times \mathbf{N}$-sorted set in which the sort of a p-ary symbol is the pair $(1, p) \in \mathbf{N} \times \mathbf{N}$. Thus, when Σ is a signature, we may form (annotated) Θ-terms with variables in Σ. For simplicity we agree that in annotated terms we omit the annotations of the symbols $\sigma \in \Sigma$. We may do so because each element of Σ has a unique sort.

Suppose now that X is an $\mathbf{N} \times \mathbf{N}$-sorted set. Then we associate a signature $\Sigma(X)$ with X defined as follows: for each $p \in \mathbf{N}$ we let $\Sigma(X)_p$ cosist of all symbols of the form $x^i_{(n,p)}$, where $n \geq 1$, $i \in [n]$, and $x \in X_{(n,p)}$.

We define two $\mathbf{N} \times \mathbf{N}$-sorted functions $\mathrm{sig}_X : X \to \mathbf{T}_\Theta(\Sigma(X))$ and $\mathrm{nosig}_X : \Sigma(X) \to$

$\mathbf{T}_\Theta(X)$ as follows:

$$\mathrm{sig}_{X(n,p)}(x) = \begin{cases} \langle x^1_{(n,p)}, x^2_{(n,p)}, \ldots, x^n_{(n,p)} \rangle & \text{if } n > 0, \\ 0_p & \text{if } n = 0, \end{cases}$$
$$\mathrm{nosig}_{X(1,p)}(x^i_{(n,p)}) = i_n \cdot x_{(n,p)},$$

for all $n, p \in \mathbf{N}$, $i \in [n]$, and $x \in X_{(n,p)}$. (Here $\langle x^1_{(n,p)}, x^2_{(n,p)}, \ldots, x^n_{(n,p)} \rangle$ and i_n are abbreviations of annotated Θ-terms defined by (3.16–3.19) on page 48.)

Note that the free extension $\mathrm{sig}^\#_X : \mathcal{T}_\Theta(X) \to \mathcal{T}_\Theta(\Sigma(X))$ of sig_X (see page 42) is the homomorphism mapping each annotated term $t \in \mathbf{T}_\Theta(X)$ to the annotated term $\mathrm{sig}^\#_X(t)$ which we get by substituting $\langle x^1_{(n,p)}, x^2_{(n,p)}, \ldots, x^n_{(n,p)} \rangle$ for $x_{(n,p)}$ and 0_p for $z_{(0,p)}$ in t, for all $n > 0$, $p \in \mathbf{N}$, $x \in X_{(n,p)}$, and $z \in X_{(0,p)}$. Similarly, for each term $t' \in \mathbf{T}_\Theta(\Sigma(X))$, $\mathrm{nosig}^\#_X(t') \in \mathbf{T}_\Theta(X)$ is the result of substituting $i_n \cdot x_{(n,p)}$ for $x^i_{(n,p)}$ in t, for all $n > 0$, $p \in \mathbf{N}$, and $x \in X_{(n,p)}$.

Although the following results are well known, we present them with proofs for the sake of completeness.

Lemma 3.2.2. *Suppose that X is an $\mathbf{N} \times \mathbf{N}$-sorted set, $\mathcal{T}$ is a preiteration theory, f is an $\mathbf{N} \times \mathbf{N}$ sorted function from X to $\mathcal{T}$, and g is an $\mathbf{N} \times \mathbf{N}$-sorted function from $\Sigma(X)$ to $\mathcal{T}$. Then $\mathrm{sig}_X \circ \mathrm{nosig}^\#_X \circ f^\# = f$ and $\mathrm{nosig}_X \circ \mathrm{sig}^\#_X \circ g^\# = g$.*

Proof. Suppose that $n, p \in \mathbf{N}$, and $x \in X_{(n,p)}$. Then we have

$$\begin{aligned} f^\#(\mathrm{nosig}^\#_X(\mathrm{sig}_X(x))) &= f^\#(\mathrm{nosig}^\#_X(\langle x^1_{(n,p)}, \ldots, x^n_{(n,p)} \rangle)) \\ &= f^\#(\langle 1_n \cdot x_{(n,p)}, \ldots, n_n \cdot x_{(n,p)} \rangle) \\ &= f^\#(x_{(n,p)}) \\ &= f(x), \end{aligned}$$

since $\mathcal{T}$ satisfies the meta-equation (3.14). This proves $\mathrm{sig}_X \circ \mathrm{nosig}^\#_X \circ f^\# = f$.

In order to prove $\mathrm{nosig}_X \circ \mathrm{sig}^\#_X \circ g^\# = g$ suppose that $x^i_{(n,p)} \in \Sigma(X)_p$, i.e., $n > 0$, $p \in \mathbf{N}$, $i \in [n]$, and $x \in X_{(n,p)}$. Then

$$\begin{aligned} g^\#(\mathrm{sig}^\#_X(\mathrm{nosig}_X(x^i_{(n,p)}))) &= g^\#(\mathrm{sig}^\#_X(i_n \cdot x_{(n,p)})) \\ &= g^\#(i_n \cdot \langle x^1_{(n,p)}, \ldots, x^n_{(n,p)} \rangle) \\ &= g^\#(x^i_{(n,p)}) \\ &= g(x^i_{(n,p)}) \end{aligned}$$

by the meta-equation (3.13). □

Theorem 3.2.3. *Suppose that $\mathcal{T}$ is a preiteration theory, X is an $\mathbf{N} \times \mathbf{N}$-sorted set of variables, and $t, t' \in \mathbf{T}_\Theta(X)$ are annotated Θ-terms with variables in X. Then $\mathcal{T}$ satisfies the annotated equation $t = t'$ if and only if the annotated equation $\mathrm{sig}^\#_X(t) = \mathrm{sig}^\#_X(t')$ holds in $\mathcal{T}$.*

Proof. First suppose that the annotated equation $t = t'$ holds in $\mathcal{T}$, and $f : \Sigma(X) \to \mathcal{T}$ is an $\mathbf{N} \times \mathbf{N}$-sorted function. Then $g := \mathrm{sig}_X \circ f^\#$ is an $\mathbf{N} \times \mathbf{N}$-sorted function from X to $\mathcal{T}$, and we have

$$f^\#(\mathrm{sig}_X^\#(t)) = g^\#(t) = g^\#(t') = f^\#(\mathrm{sig}_X^\#(t')),$$

by Lemma 3.1.1. This proves $\mathcal{T} \models \mathrm{sig}_X^\#(t) = \mathrm{sig}_X^\#(t')$.

Now suppose that $\mathcal{T} \models \mathrm{sig}_X^\#(t) = \mathrm{sig}_X^\#(t')$ and $g : X \to \mathcal{T}$ is an $\mathbf{N} \times \mathbf{N}$-sorted function. Then $f := \mathrm{nosig}_X \circ g^\#$ is an $\mathbf{N} \times \mathbf{N}$-sorted function from $\Sigma(X)$ to $\mathcal{T}$, and we have

$$\begin{aligned} g^\# &= (\mathrm{sig}_X \circ \mathrm{nosig}_X^\# \circ g^\#)^\# && \text{(by Lemma 3.2.2)} \\ &= \mathrm{sig}_X^\# \circ \mathrm{nosig}_X^\# \circ g^\# && \text{(by Lemma 3.1.1)} \\ &= \mathrm{sig}_X^\# \circ f^\#. && \text{(by Lemma 3.1.1)} \end{aligned}$$

It follows that $g^\#(t) = f^\#(\mathrm{sig}_X^\#(t)) = f^\#(\mathrm{sig}_X^\#(t')) = g^\#(t')$ proving $\mathcal{T} \models t = t'$. □

Theorem 3.2.4. *Suppose that $\mathcal{V}$ is a variety of preiteration theories and X is an $\mathbf{N} \times \mathbf{N}$-sorted set. Then the X-generated free algebra in $\mathcal{V}$ is isomorphic to the $\Sigma(X)$-generated free algebra in $\mathcal{V}$.*

Proof. Let $f : X \to \mathcal{F}_\mathcal{V}(\Sigma(X))$ be the $\mathbf{N} \times \mathbf{N}$-sorted function $\mathrm{sig}_X \circ (\eta^\mathcal{V}_{\Sigma(X)})^\#$, and $g : \Sigma(X) \to \mathcal{F}_\mathcal{V}(X)$ the $\mathbf{N} \times \mathbf{N}$-sorted function $\mathrm{nosig}_X \circ (\eta^\mathcal{V}_X)^\#$. Then we have

$$\begin{aligned} \eta^\mathcal{V}_X \circ f^{\#\mathcal{V}} \circ g^{\#\mathcal{V}} &= \mathrm{sig}_X \circ (\eta^\mathcal{V}_{\Sigma(X)})^\# \circ g^{\#\mathcal{V}} && \text{(since } \eta^\mathcal{V}_X \circ f^{\#\mathcal{V}} = f\text{)} \\ &= \mathrm{sig}_X \circ (\eta^\mathcal{V}_{\Sigma(X)} \circ g^{\#\mathcal{V}})^\# && \text{(by Lemma 3.1.1)} \\ &= \mathrm{sig}_X \circ (\mathrm{nosig}_X \circ (\eta^\mathcal{V}_X)^\#)^\# && \text{(since } \eta^\mathcal{V}_{\Sigma(X)} \circ g^{\#\mathcal{V}} = g\text{)} \\ &= \mathrm{sig}_X \circ \mathrm{nosig}_X^\# \circ (\eta^\mathcal{V}_X)^\# && \text{(by Lemma 3.1.1)} \\ &= \eta^\mathcal{V}_X. && \text{(by Lemma 3.2.2)} \end{aligned}$$

Since $(\eta^\mathcal{V}_X)^{\#\mathcal{V}} = \mathrm{id}_{\mathrm{F}_\mathcal{V}(X)}$ is the only homomorphism $h : \mathcal{F}_\mathcal{V}(X) \to \mathcal{F}_\mathcal{V}(X)$ with $\eta^\mathcal{V}_X \circ h = \eta^\mathcal{V}_X$, it follows that $f^{\#\mathcal{V}} \circ g^{\#\mathcal{V}} = \mathrm{id}_{\mathrm{F}_\mathcal{V}(X)}$. A similar argument shows that $g^{\#\mathcal{V}} \circ f^{\#\mathcal{V}} = \mathrm{id}_{\mathrm{F}_\mathcal{V}(\Sigma(X))}$. Thus, $f^{\#\mathcal{V}}$ is an isomorphism from $\mathcal{F}_\mathcal{V}(X)$ to $\mathcal{F}_\mathcal{V}(\Sigma(X))$. □

In particular, Theorem 3.2.4 applies to the variety of iteration theories and the variety of Conway theories, so that if we want to describe all free Conway (iteration) theories then it is sufficient to describe those which are generated by a signature Σ.

3.3 Algebras of flowchart schemes

As a first step towards describing the free Conway theories, in this section we define the $\mathbf{N} \times \mathbf{N}$-sorted Θ-algebras of flowchart schemes. Many variants of flowcharts

schemes and algebras thereof were defined and investigated by many authors. Our basic references are [25, 27, 13], see also [28, 19, 48, 5, 1, 2, 3, 4].

Throughout this section Σ denotes a fixed signature, that is, an $\mathbf{N}$-labeled set. (See page 12 for the definition of labeled sets.)

3.3.1 Σ-schemes

A *(deterministic) Σ-scheme of sort* $(n,p) \in \mathbf{N} \times \mathbf{N}$ is a finite labeled digraph $\mathcal{S}$ satisfying the following conditions. The nodes of $\mathcal{S}$ are partitioned into three disjoint subsets: *input nodes* $\mathrm{in}_1, \ldots, \mathrm{in}_n$, *output nodes* $\mathrm{out}_1, \ldots, \mathrm{out}_p$, and *states*. Input nodes have in-degree 0, and output nodes have out-degree 0. The out-degree of input nodes is at most 1. Each state s of $\mathcal{S}$ is labeled by a symbol $\sigma \in \Sigma$. The outgoing edges of a state s having label $\sigma \in \Sigma_n$ are labeled by distinct integers between 1 and n.

Let $\mathbf{I} = \{\mathrm{in}_1, \mathrm{in}_2, \mathrm{in}_3, \ldots\}$ and $\mathbf{O} = \{\mathrm{out}_1, \mathrm{out}_2, \mathrm{out}_3, \ldots\}$ be the set of all input and output nodes, respectively. We shall write $\mathbf{I}_n$ for $\{\mathrm{in}_1, \ldots, \mathrm{in}_n\}$ and $\mathbf{O}_p$ for $\{\mathrm{out}_1, \ldots, \mathrm{out}_p\}$. Then a Σ-scheme of sort (n,p) may be represented as a 4-tuple $\mathcal{S} = (S, \alpha_\mathcal{S}, \delta_\mathcal{S}, n, p)$, where

- S is the finite Σ-labeled set of states such that $S \cap \mathbf{I} = S \cap \mathbf{O} = \emptyset$,
- $\alpha_\mathcal{S} : \mathbf{I}_n \to S \cup \mathbf{O}_p$ is the partial *input function*,
- $\delta_\mathcal{S} : S \times [\omega] \to S \cup \mathbf{O}_p$ is the partial *transition function* satisfying

$$(s,i) \in \mathrm{Dom}(\delta) \ \wedge \ \mathrm{label}_\mathcal{S}(s) \in \Sigma_n \Longrightarrow i \leq n.$$

- $n \in \mathbf{N}$ is the *source* of $\mathcal{S}$,
- $p \in \mathbf{N}$ is the *target* of $\mathcal{S}$.

We usually write $\mathcal{S} : n \to p$ to indicate that $\mathcal{S}$ has source n and target p, that is, $\mathcal{S}$ is of sort (n,p).

When it is clear from the context which scheme $\mathcal{S}$ we are talking about we shall simply write α and δ instead of $\alpha_\mathcal{S}$ and $\delta_\mathcal{S}$. We usually denote schemes by calligraphic letters, and their set of states by the corresponding capital letters.

We say that two Σ-schemes $\mathcal{S}$ and $\mathcal{S}'$ are *isomorphic* if they are isomorphic as labeled directed graphs. We are more interested in *abstract Σ-schemes*, (that is, isomorphisms classes of Σ-schemes) than Σ-schemes themselves. Even so, in order to make our presentation simpler, we shall work with Σ-schemes, but *we shall consider isomorphic Σ-schemes to be identical.* Due to this convention, when needed, we may assume without loss of generality that any two schemes have disjoint sets of states.

Suppose that $\mathcal{S} : n \to p$ is a Σ-scheme. Then each word $u \in [\omega]^*$ induces a partial function $u_\mathcal{S} : \mathbf{I}_n \cup S \to S \cup \mathbf{O}_p$ defined by

$$u_\mathcal{S} = \begin{cases} \alpha \cup \mathrm{id}_S & \text{if } u = \epsilon, \\ v_\mathcal{S} \circ \delta_i & \text{if } u = vi,\ v \in [\omega]^*,\ i \in [\omega], \end{cases}$$

where $\delta_i : S \to S \cup \mathbf{O}_p$ is the partial function defined by

$$\delta_i(s) = y \iff \delta(s, i) = y,$$

for all $s \in S$ and $y \in S \cup \mathbf{O}_p$.

Intuitively, $x\, u_\mathcal{S}\, y$ ($x \in \mathbf{I}_n \cup S$, $y \in S \cup \mathbf{O}_p$) means that there is a directed u-labeled walk in the graph of $\mathcal{S}$ from node x to node y.[1] In order to visualize this intuitive meaning we shall sometimes write $x \xrightarrow{u}_\mathcal{S} y$ instead of $x\, u_\mathcal{S}\, y$, or simply $x \xrightarrow{u} y$, when $\mathcal{S}$ is understood.

Suppose that $u \in [\omega]^*$, $C \subseteq S \cup \mathbf{I}_n$, and $D \subseteq S \cup \mathbf{O}_p$. Then we denote by $u_\mathcal{S}[C, D]$ the restriction of the relation induced by the word u in scheme $\mathcal{S}$ to the set $C \times D$, that is, we define

$$u_\mathcal{S}[C, D] := u_\mathcal{S} \cap (C \times D).$$

Note that $u_\mathcal{S}[C, D]$ is a partial function from C to D. The collection of all *nonempty* partial functions from C to D induced by the words in $[\omega]^*$ is denoted $[\omega]^*_\mathcal{S}[C, D]$, i.e.,

$$[\omega]^*_\mathcal{S}[C, D] = \{u_\mathcal{S}[C, D] \mid u \in [\omega]^*\} \setminus \{\emptyset\}.$$

3.3.2 Base schemes

Suppose that $\mathcal{S} : n \to p$ is a Σ-scheme. We say that $\mathcal{S}$ is a *partial base scheme* if $\mathcal{S}$ has no states. If in addition $\alpha_\mathcal{S}$ is a total function from $\mathbf{I}_n$ to $\mathbf{O}_p$ then we say that $\mathcal{S}$ is a *base scheme*.

Observe that there is a bijective correspondence between (partial) base schemes $\mathcal{S} : n \to p$ and (partial) functions $f : [n] \to [p]$. Each (partial) base scheme $\mathcal{S} : n \to p$ determines a (partial) function $f : [n] \to [p]$ defined by

$$f(i) = j \iff \alpha(\mathrm{in}_i) = \mathrm{out}_j, \qquad (*)$$

for all $i \in [n]$ and $j \in [p]$, and conversely, if $f : [n] \to [p]$ is a (partial) function then there is a unique (partial) base scheme $\mathcal{S} : n \to p$ such that $(*)$ holds.

[1] The label of a walk is the concatenation of the labels of the edges appearing in it. According to our definition, the outgoing edges of input nodes have no label. Here we consider them to be labeled by the empty word ϵ.

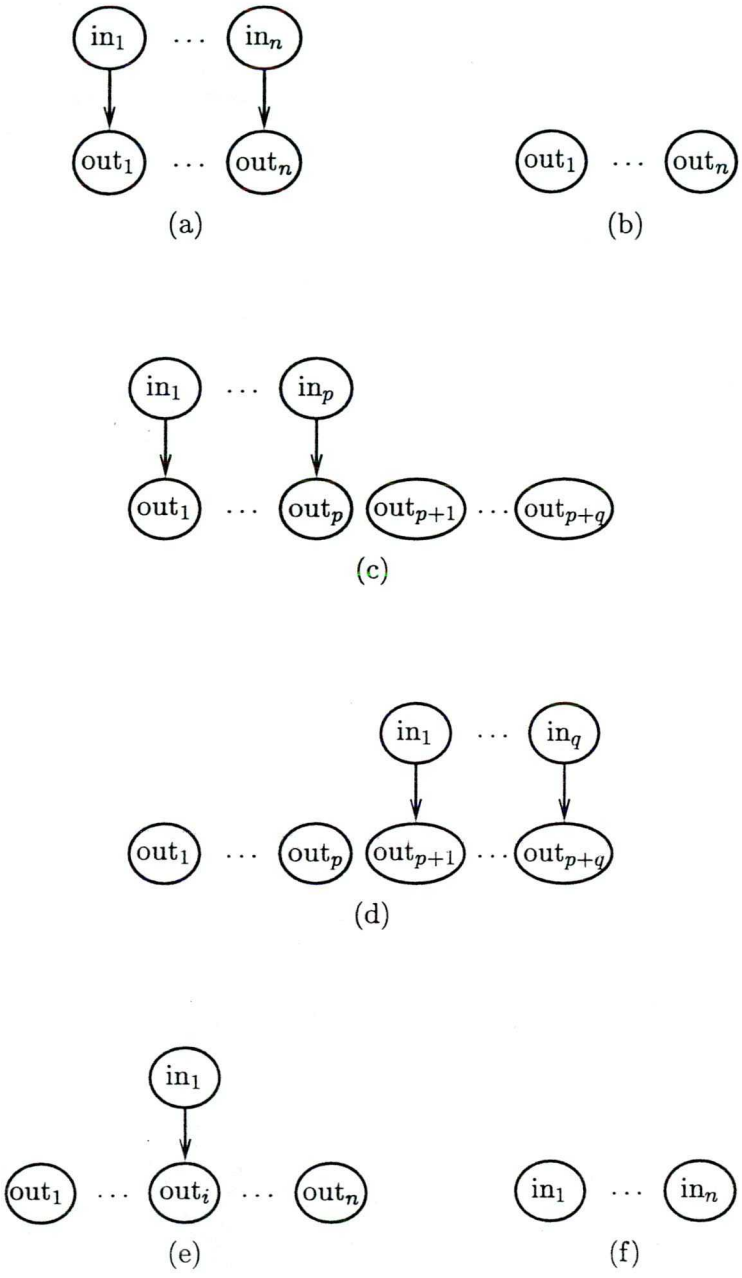


Figure 3.1: (Partial) base schemes

3.3.3 The Θ-algebra of Σ-schemes

Since each Σ-scheme has a unique sort $(n,p) \in \mathbf{N} \times \mathbf{N}$, and since isomorphic Σ-schemes are necessarily of the same sort, abstract Σ-schemes form an $\mathbf{N} \times \mathbf{N}$-labeled set Sch_Σ. We promote this set to a Θ-algebra $\mathcal{S}ch_\Sigma$ by defining the constants and operations as follows.

Suppose that n, p, q are nonnegative integers. Then

$\mathbf{1}_n$ is the base scheme corresponding to the identity function $\mathrm{id}_{[n]} : [n] \to [n]$, see Figure 3.1(a),

0_n is the unique base scheme $0 \to n$, see Figure 3.1(b),

$\kappa_{p,q}$ is the base scheme corresponding to the inclusion $[p] \to [p+q]$, $x \mapsto x$, see Figure 3.1(c),

$\lambda_{p,q}$ is the base scheme corresponding to the translated inclusion $[q] \to [p+q]$, $x \mapsto p + x$, see Figure 3.1(d).

The composition, pairing, and iteration operations are defined as follows.

Pairing: Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : m \to p$ are Σ-schemes with $S \cap S' = \emptyset$. Then $\langle \mathcal{S}, \mathcal{S}' \rangle : n + m \to p$ is the Σ-scheme having states $S \cup S'$, input function $\alpha_{\mathcal{S}} \cup (\mathrm{ishift}_n^{-1} \circ \alpha_{\mathcal{S}'})$, and transition function $\delta_{\mathcal{S}} \cup \delta_{\mathcal{S}'}$, where $\mathrm{ishift}_n : \mathbf{I} \to \mathbf{I}$ is the function mapping each input node in_i to in_{n+i}.

The graph of $\langle \mathcal{S}, \mathcal{S}' \rangle$ can be constructed from the graphs of $\mathcal{S}$ and $\mathcal{S}'$ as follows: first replace each node in_i with in_{n+i} in the graph of $\mathcal{S}'$, then take the disjoint union of this graph with the graph of $\mathcal{S}$, and identify the corresponding output nodes of $\mathcal{S}$ and $\mathcal{S}'$. See Figure 3.2.

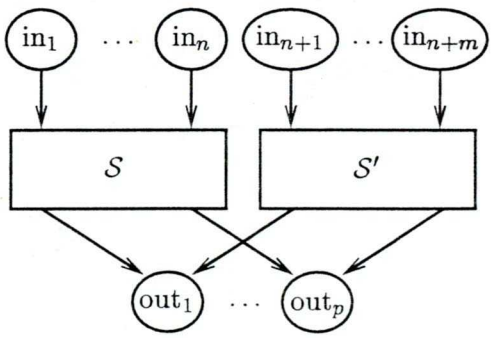


Figure 3.2: Pairing

Composition: Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : p \to q$ are Σ-schemes with $S \cap S' = \emptyset$. The composition operation applied to $\mathcal{S}$ and $\mathcal{S}'$ yields a Σ-scheme $\mathcal{S} \cdot \mathcal{S}' : n \to q$ with states $S \cup S'$, input function $\alpha_{\mathcal{S}} \circ (\mathrm{id}_S \cup \mathrm{oi} \circ \alpha_{\mathcal{S}'})$, and

transition function $\delta_{\mathcal{S}} \circ (\mathrm{id}_S \cup \mathrm{oi} \circ \alpha_{\mathcal{S}'}) \cup \delta_{\mathcal{S}'}$, where $\mathrm{oi} : \mathbf{O} \to \mathbf{I}$ is the function mapping each output node out_i to the corresponding input node in_i.

The graph of $\mathcal{S} \cdot \mathcal{S}'$ can be constructed from the graph of $\mathcal{S}$ and $\mathcal{S}'$ in the following way: delete the output nodes of $\mathcal{S}$ and the input nodes of $\mathcal{S}'$ together with all adjacent edges, then take the disjoint union of the two graphs, and draw a new edge from state $s \in S$ to state $s' \in S'$ labeled $t \in [\omega]$ if and only if there exists some $j \in [p]$ such that $s \xrightarrow{t} \mathrm{out}_j$ in $\mathcal{S}$, and $\mathrm{in}_j \to s'$ in $\mathcal{S}'$. See Figure 3.3.

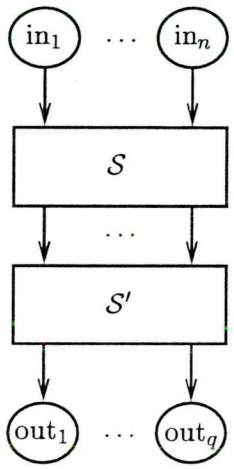


Figure 3.3: Composition

Iteration: Suppose that $\mathcal{S} : n \to n+p$ is a Σ-scheme. Then the iteration operation applied to $\mathcal{S}$ yields a Σ-scheme $\mathcal{S}^\dagger$ having the same states as $\mathcal{S}$, input function $\alpha_{\mathcal{S}} \circ (\mathrm{id}_S \cup (\mathrm{oi} \circ \alpha_{\mathcal{S}})^* \circ \mathrm{oshift}_n^{-1})$, and transition function $\delta_{\mathcal{S}} \circ (\mathrm{id}_S \cup (\mathrm{oi} \circ \alpha_{\mathcal{S}})^* \circ \mathrm{oshift}_n^{-1})$, where $\mathrm{oshift}_n : \mathbf{O} \to \mathbf{O}$ is the function mapping each output node out_i to the output node out_{n+i}.

The graph of $\mathcal{S}^\dagger$ is constructed from the graph of $\mathcal{S}$ as follows: for each pair $x \in \mathbf{I}_n \cup S$, $y \in S \cup \mathbf{O}_{n+p}$ of nodes, draw a new edge from x to y labeled $t \in \mathbf{N} \cup \{\epsilon\}$ whenever there exist indices $i_1, \ldots, i_m \in [n]$ $(m \geq 1)$ such that

$$x \xrightarrow{t} \mathrm{out}_{i_1}, \quad \mathrm{in}_{i_1} \xrightarrow{\epsilon} \mathrm{out}_{i_2}, \quad \mathrm{in}_{i_2} \xrightarrow{\epsilon} \mathrm{out}_{i_3}, \quad \ldots, \quad \mathrm{in}_{i_m} \xrightarrow{\epsilon} y$$

in $\mathcal{S}$. Then delete the first n output nodes together with all adjacent edges, and replace each output node out_{n+i} $(i \in [p])$ with out_i. See Figure 3.4. Note that Elgot only defined a partial † operation. The † operation was made total in [13].

Recall that in any Θ-algebra we define the constants $i_n : 1 \to n$ $(n \in \mathbf{N})$ by equation (3.19). The reader may easily verify that i_n is the base scheme corresponding to the map $1 \mapsto i$, see Figure 3.1(e).

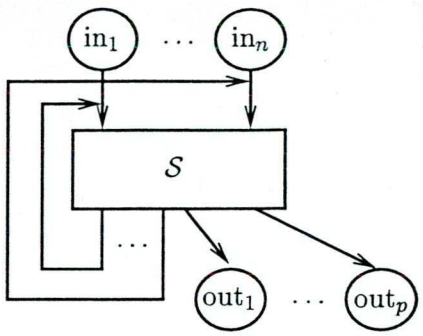


Figure 3.4: Iteration

We shall also use the partial base schemes $\bot_n : n \to 0$ ($n \in \mathbf{N}$), see Figure 3.1(f). Note that $\bot_n = \mathbf{1}_n^\dagger$ is the unique partial base scheme of sort $n \to 0$.

The separated sum operation on Σ-schemes is defined by (3.7). The reader may verify that the separated sum of two Σ-schemes $\mathcal{S} : n \to p$ and $\mathcal{S}' : m \to q$ ($S \cap S' = \emptyset$) is the Σ-scheme $\mathcal{S} \oplus \mathcal{S}' : n+m \to p+q$ with states $S \cup S'$, input function $\alpha_{\mathcal{S}} \cup \mathrm{ishift}_n^{-1} \circ \alpha_{\mathcal{S}'}$ and transition function $\delta_{\mathcal{S}} \cup \delta_{\mathcal{S}'} \circ \mathrm{oshift}_p$.

The graph of $\mathcal{S} \oplus \mathcal{S}'$ is constructed as follows: replace each input node in_i with in_{n+i} and each output node out_j with out_{p+j} in the graph of $\mathcal{S}'$, then take the disjoint union of the two graphs. See Figure 3.5.

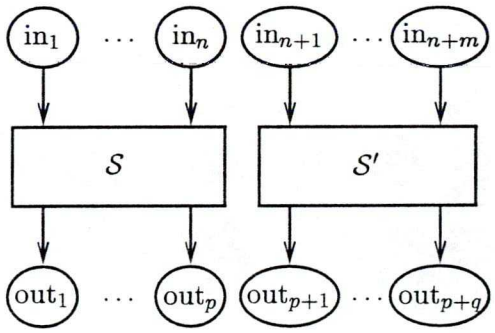


Figure 3.5: Separated sum

With each symbol $\sigma \in \Sigma_p$ we associate an *atomic* Σ*-scheme* $\widehat{\sigma} : 1 \to p$ having a unique state labeled σ, see Figure 3.6. It is not hard to see that the algebra $\mathcal{S}ch_\Sigma$ is generated by the $\mathbf{N} \times \mathbf{N}$-labeled set of atomic Σ-schemes. In the sequel we shall identify each atomic scheme $\widehat{\sigma} : 1 \to p$ with the symbol $\sigma \in \Sigma_p$, and we shall simply say that $\mathcal{S}ch_\Sigma$ is generated by the signature Σ. In fact, each Σ-scheme $\mathcal{S} : n \to p$ can be written as

$$\rho \cdot \langle \langle \widehat{\sigma_1} \cdot \rho_1, \ldots, \widehat{\sigma_m} \cdot \rho_m \rangle^\dagger, \mathbf{1}_p \rangle$$

for some partial base scheme $\rho : n \to m+p$, atomic schemes $\widehat{\sigma_i} : 1 \to p_i$ and partial

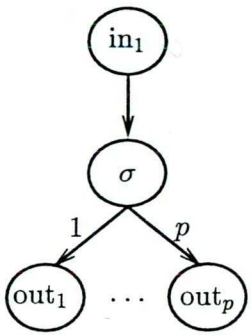


Figure 3.6: The atomic scheme $\widehat{\sigma} : 1 \to p$

base schemes $\rho_i : p_i \to m+p$, $i \in [m]$, where $\sigma_i \in \Sigma_{p_i}$, and m is the number of states in $\mathcal{S}$. See [27]. Each partial base scheme $\rho : n \to p$ can be expressed uniquely as an n-tuple of schemes of the form $i_p : 1 \to p$ or $1_1^\dagger \cdot 0_p$.

Observe that if Σ is empty then each element of $\mathcal{S}ch_\Sigma$ is a partial base scheme, and $\mathcal{S}ch_\Sigma$ is isomorphic to the Θ-algebra **Pfn** of all partial functions $[n] \to [p]$, $n, p \in \mathbf{N}$. Since **Pfn** is freely generated by the empty set in the class of all Conway theories, so is $\mathcal{S}ch_\emptyset$.

However, if Σ is not empty then the Θ-algebra $\mathcal{S}ch_\Sigma$ is not even a preiteration theory, since it does not satisfy the identities (3.31), (3.32) and (3.33). Interestingly, $\mathcal{S}ch_\Sigma$ satisfies two of the Conway identities, namely, the parameter and double dagger identities, and it also satisfies the composition identity (3.48) in the special case when f is a partial base scheme. Thus, $\mathcal{S}ch_\Sigma$ satisfies the following (infinite) collection of meta equations, the so called

BASE COMPOSITION IDENTITIES:

$$(\phi \cdot \langle g, \lambda \rangle)^\dagger \approx \phi \cdot \langle (g \cdot \langle \phi, \lambda \rangle)^\dagger, \ \mathbf{1} \rangle,$$

for all partial base terms ϕ.

Definition 3.3.1. Suppose that Δ is an $\mathbf{N}$-labeled set. Then we denote by $\equiv_\Delta$ the least congruence on $\mathcal{S}ch_\Delta$ such that the quotient $\mathcal{S}ch_\Delta/\equiv_\Delta$ satisfies the meta-equations (3.33) and (3.34). When the signature Δ is clear from the context, we shall simply write $\equiv$ for $\equiv_\Delta$.

Lemma 3.3.1. *Suppose that $\mathcal{F} : n \to 2n+m+p$ and $\mathcal{G} : m \to 2n+m+p$ are Σ-schemes. Then*

$$\langle \langle \mathcal{F}, \mathcal{F} \rangle, \mathcal{G} \rangle^\dagger \equiv \beta \cdot (\langle \mathcal{F}, \mathcal{G} \rangle \cdot (\beta \oplus \mathbf{1}_p))^\dagger,$$

where β denotes the base scheme $\langle \mathbf{1}_n, \mathbf{1}_n \rangle \oplus \mathbf{1}_m : 2n+m \to n+m$.

Proof. By the definition of $\equiv$.

$$\langle \mathbf{1}_n, \mathbf{1}_n \rangle \cdot \mathcal{F} \ \equiv \ \langle \kappa_{n,n} \cdot \langle \mathbf{1}_n, \mathbf{1}_n \rangle \cdot \mathcal{F}, \ \lambda_{n,n} \cdot \langle \mathbf{1}_n, \mathbf{1}_n \rangle \cdot \mathcal{F} \rangle \ = \ \langle \mathcal{F}, \mathcal{F} \rangle.$$

It follows that

$$\beta \cdot \langle \mathcal{F}, \mathcal{G} \rangle \;=\; \langle \langle \mathbf{1}_n, \mathbf{1}_n \rangle \cdot \mathcal{F}, \mathcal{G} \rangle \;\equiv\; \langle \langle \mathcal{F}, \mathcal{F} \rangle, \mathcal{G} \rangle,$$

and thus

$$\langle \langle \mathcal{F}, \mathcal{F} \rangle, \mathcal{G} \rangle^\dagger \;\equiv\; (\beta \cdot \langle \mathcal{F}, \mathcal{G} \rangle)^\dagger \;=\; \beta \cdot (\langle \mathcal{F}, \mathcal{G} \rangle \cdot (\beta \oplus \mathbf{1}_p))^\dagger,$$

where the last equality comes from the composition identity applied to the base scheme $\beta \oplus 0_p : 2n+m \to n+m+p$ and the Σ-scheme $\langle \mathcal{F}, \mathcal{G} \rangle : n+m \to 2n+m+p$. □

Theorem 3.3.2. *$Sch_\Sigma/\equiv$ is freely generated by the signature Σ in the variety of all Conway theories.*

Proof. It is known that Sch_Σ is freely generated by Σ in the smallest variety containing the Θ-algebras Sch_Δ, for all signatures Δ. A complete axiomatization of this variety was given in [13]. Since each of those axioms is a logical consequence of the Conway identities, it follows that the Σ-generated free Conway theory is the quotient $Sch_\Sigma/\sim$, where $\sim$ is the least congruence on Sch_Σ for which $Sch_\Sigma/\sim$ is a Conway theory. We are going to show that $\equiv \;=\; \sim$.

Since $Sch_\Sigma/\sim$ satisfies the meta-equations (3.33–3.34), we have $\equiv \subseteq \sim$. The converse containment $\equiv \supseteq \sim$ is proved by showing that $Sch_\Sigma/\equiv$ is a Conway theory. Except for the composition identity and the preiteration theory identities (3.31–3.34), all defining axioms of Conway theories hold in Sch_Σ, and hence in the quotient $Sch_\Sigma/\equiv$. As (3.33) and (3.34) hold in $Sch_\Sigma/\equiv$ by definition, we are left to show that $Sch_\Sigma/\equiv$ satisfies (3.31), (3.32) and the composition identity.

In order to prove that (3.31) and (3.32) hold in $Sch_\Sigma/\equiv$ suppose that $\mathcal{F} : n \to p$ and $G : m \to p$ are two Σ-schemes. Then we have

$$\kappa_{n,m} \cdot \langle \mathcal{F}, \mathcal{G} \rangle \;=\; \langle \mathcal{F}, 0_m \cdot \mathcal{G} \rangle \;\equiv\; \langle \mathcal{F}, 0_p \rangle \;=\; \mathcal{F},$$

and

$$\lambda_{n,m} \cdot \langle \mathcal{F}, \mathcal{G} \rangle \;=\; \langle 0_n \cdot \mathcal{F}, \mathcal{G} \rangle \;\equiv\; \langle 0_p, \mathcal{G} \rangle \;=\; \mathcal{G}.$$

Now suppose that $\mathcal{F} : n \to m+p$ and $\mathcal{G} : m \to n+p$ are Σ-schemes and let β denote the base scheme $\langle \mathbf{1}_n, \mathbf{1}_n \rangle \oplus \mathbf{1}_m : 2n + m \to n + m$. Then

$$\begin{aligned}
&\mathcal{F} \cdot \langle (\mathcal{G} \cdot \langle \mathcal{F}, \lambda_{m,p} \rangle)^\dagger, \mathbf{1}_p \rangle \\
&= \kappa_{n,n+m} \cdot \langle \langle \mathcal{F} \cdot \lambda_{2n,m+p},\ \mathcal{F} \cdot \lambda_{2n,m+p} \rangle,\ \mathcal{G} \cdot (\lambda_{n,n} \oplus \lambda_{m,p}) \rangle^\dagger \\
&\equiv \kappa_{n,n+m} \cdot \beta \cdot (\langle \mathcal{F} \cdot \lambda_{2n,m+p},\ \mathcal{G} \cdot (\lambda_{n,n} \oplus \lambda_{m,p}) \rangle \cdot (\beta \oplus \mathbf{1}_p))^\dagger \\
&= \kappa_{n,m} \cdot \langle \mathcal{F} \cdot \lambda_{n,m+p},\ \mathcal{G} \cdot (\mathbf{1}_n \oplus \lambda_{m,p}) \rangle^\dagger \\
&= (\mathcal{F} \cdot \langle \mathcal{G}, \lambda_{n,p} \rangle)^\dagger,
\end{aligned}$$

by Lemma 3.3.1. □

Chapter 4

The free Conway theories

In Theorem 3.3.2 we have given a description of the free Conway theories generated by a signature Σ: it is the quotient Θ-algebra $\mathcal{S}ch_\Sigma/\equiv$, where $\equiv$ is the least congruence on $\mathcal{S}ch_\Sigma$ such that $\mathcal{S}ch_\Sigma/\equiv$ satisfies the meta-equations (3.33) and (3.34). This description cannot be called concrete until we know how to decide the *Conway equivalence problem* of Σ-schemes, that is, how to decide if $\mathcal{S} \equiv \mathcal{S}'$ holds for two Σ-schemes $\mathcal{S}$ and $\mathcal{S}'$. We present an algorithm for this decision problem based on Theorem 4.1.27 and Theorem 4.3.2. Then we prove in Theorem 4.3.3 that the Conway equivalence problem is **PSPACE**-complete, as well as the equational theory of Conway theories, and some other related decision problems.

Lemma 3.3.1, Theorem 3.3.2, and the results contained in this chapter were published in [8].

Throughout this chapter Σ denotes a fixed signature. We remind the reader that some of the notations used here were defined in Chapter 1.

4.1 Simulations

In this section we define simulations of Σ-schemes, i.e., structure preserving relations between Σ-schemes. Congruences and homomorphisms of Σ-schemes are then defined as simulations satisfying some further requirements.

Definition 4.1.1. Suppose that $\mathcal{S}$ and $\mathcal{S}'$ are Σ-schemes $n \to p$. A Σ-sorted relation $\gamma \subseteq S \times S'$ is called a *simulation from $\mathcal{S}$ to $\mathcal{S}'$*, written $\mathcal{S} <\gamma> \mathcal{S}'$, if

$$(\mathrm{in}_i\, \alpha)\ (\gamma \cup \mathrm{id}_{\mathbf{O}})\ (\mathrm{in}_i\, \alpha') \tag{4.1}$$

and

$$s\ \gamma\ s' \Longrightarrow (s\, t_S)\ (\gamma \cup \mathrm{id}_{\mathbf{O}})\ (s'\, t_{S'}) \tag{4.2}$$

hold for all $i \in [n]$, $s \in S$, $s' \in S'$ and $t \in [\omega]$. We write $\mathcal{S} \approx \mathcal{S}'$ and say that the two schemes $\mathcal{S}$ and $\mathcal{S}'$ are *strongly equivalent* if there exists a simulation from $\mathcal{S}$ to $\mathcal{S}'$.

In the special case that the simulation relation γ is a function $S \to S'$, γ is called a *homomorphism* from $\mathcal{S}$ to $\mathcal{S}'$. A bijective homomorphism is called an *isomorphism*. Another special case is that $\mathcal{S} = \mathcal{S}'$ and the simulation γ is an equivalence relation on S: then we say γ is a *congruence* on $\mathcal{S}$.

Note that if γ is a simulation from $\mathcal{S}$ to $\mathcal{S}'$ then the following hold for the graphical representations of $\mathcal{S}$ and $\mathcal{S}'$. First, for any $i \in [n]$, the ith input node of $\mathcal{S}$ has an out-edge if and only if the ith input node of $\mathcal{S}'$ has an out-edge. Moreover, if the ith input node of $\mathcal{S}$ is connected by an edge to an internal node s, then the ith input node of $\mathcal{S}'$ is connected by an edge to an internal node s' with $(s, s') \in \gamma$. If the ith input node of $\mathcal{S}$ is connected to an output node, then the ith input node of $\mathcal{S}'$ is connected to the corresponding output node of $\mathcal{S}'$. And if $s \in S$ and $s' \in S'$ with $(s, s') \in \gamma$, then s and s' have the same label, and for any $t \in [\omega]$, s has an out-edge labeled by t if and only if s' has one. Moreover, if the target of the out-edge of s labeled by t is an internal node v, then so is the target v' of the corresponding out-edge of s', and $(v, v') \in \gamma$. If the target of the out-edge of s labeled by t is an output node, then the target of the out-edge of s' labeled by t is the corresponding output node of $\mathcal{S}'$.

We usually write $\gamma : \mathcal{S} \to \mathcal{S}'$ to indicate that γ is a homomorphism from $\mathcal{S}$ to $\mathcal{S}'$. Simulations have several nice properties, some of them are listed in the following lemma. See also [48, 19, 15].

Lemma 4.1.1. *For all relations φ, ψ, and all Σ-schemes $\mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{F}', \mathcal{G}'$ of appropriate source and target,*

1. $\mathcal{F} <\mathrm{id}_F> \mathcal{F}$
2. $\mathcal{F} <\varphi> \mathcal{G} \implies \mathcal{G} <\varphi^{-1}> \mathcal{F}$
3. $\mathcal{F} <\varphi> \mathcal{G} \wedge \mathcal{G} <\psi> \mathcal{H} \implies \mathcal{F} <\varphi \circ \psi> \mathcal{H}$
4. $\mathcal{F} <\varphi> \mathcal{F}' \wedge \mathcal{G} <\psi> \mathcal{G}' \implies \langle \mathcal{F}, \mathcal{G} \rangle <\varphi \cup \psi> \langle \mathcal{F}', \mathcal{G}' \rangle$
5. $\mathcal{F} <\varphi> \mathcal{F}' \wedge \mathcal{G} <\psi> \mathcal{G}' \implies \mathcal{F} \cdot \mathcal{G} <\varphi \cup \psi> \mathcal{F}' \cdot \mathcal{G}'$
6. $\mathcal{F} <\varphi> \mathcal{F}' \wedge \mathcal{G} <\psi> \mathcal{G}' \implies \mathcal{F} \div \mathcal{G} <\varphi \cup \psi> \mathcal{F}' \oplus \mathcal{G}'$
7. $\mathcal{F} <\varphi> \mathcal{G} \implies \mathcal{F}^\dagger <\varphi> \mathcal{G}^\dagger$
8. $\mathcal{F} <\varphi> \mathcal{G} \wedge \mathcal{F} <\psi> \mathcal{G} \implies \mathcal{F} <\varphi \cup \psi> \mathcal{G}$
9. $\mathcal{F} <\varphi> \mathcal{G} \wedge \mathcal{F} <\psi> \mathcal{G} \implies \mathcal{F} <\varphi \cap \psi> \mathcal{G}$

□

Corollary 4.1.2. *The strong equivalence relation $\approx$ is a congruence on the Θ-algebra Sch_Σ. Moreover, when $\mathcal{S}$ and $\mathcal{S}'$ are strongly equivalent schemes, there exists a smallest simulation*

$$S\Gamma_{S'} := \bigcap_{\mathcal{S} <\gamma> \mathcal{S}'} \gamma$$

and a largest simulation

$$ {}_{\mathcal{S}}\Omega_{\mathcal{S}'} := \bigcup_{\mathcal{S} <\gamma> \mathcal{S}'} \gamma $$

from $\mathcal{S}$ to $\mathcal{S}'$. □

Lemma 4.1.3. *Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : n \to p$ are strongly equivalent Σ-schemes. Then*

$$ s \; {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \; s' \iff \exists i \in [n] \; \exists u \in [\omega]^* \; s = u_{\mathcal{S}}(\mathrm{in}_i) \wedge s' = u_{\mathcal{S}'}(\mathrm{in}_i) $$

and

$$ s \; {}_{\mathcal{S}}\Omega_{\mathcal{S}'} \; s' \iff \forall u \in [\omega]^* \; s\, u_{\mathcal{S}} \, (\mathrm{label}_{\mathcal{S}} \cup \mathrm{id}_{\mathbf{O}}) = s'\, u_{\mathcal{S}'} \, (\mathrm{label}_{\mathcal{S}'} \cup \mathrm{id}_{\mathbf{O}}), $$

for all $s \in S$ and $s' \in S'$. Moreover, ${}_{\mathcal{S}}\Omega_{\mathcal{S}}$ is the largest congruence on $\mathcal{S}$. □

We shall write $\Theta_{\mathcal{S}}$ for ${}_{\mathcal{S}}\Omega_{\mathcal{S}}$.

Thus, two states $s \in S$ and $s' \in S'$ are related by the smallest simulation if and only if there exist a word $u \in [\omega]^*$ and an integer $i \in [n]$ such that s is the target of the directed walk, labeled by u, from node $\alpha(i)$ of $\mathcal{S}$, and s' is the target of the directed walk from $\alpha'(i)$ labeled by the same word u. Moreover, s is related to s' by the largest simulation if and only if for all words $u \in [\omega]^*$ there is a directed walk labeled by u from s if and only if there is a directed walk labeled by u from s', and the labels of the targets of these walks agree.

Definition 4.1.2. Suppose that $\mathcal{S} : n \to p$ is a Σ-scheme and ρ is a congruence on $\mathcal{S}$. Then the *quotient of $\mathcal{S}$ with respect to ρ* is the Σ-scheme $\mathcal{S}/\rho = (S/\rho, \alpha_{\mathcal{S}/\rho}, \delta_{\mathcal{S}/\rho}) : n \to p$, where $\alpha_{\mathcal{S}/\rho}$ and $\delta_{\mathcal{S}/\rho}$ are defined by

$$ \mathrm{in}_i \, \alpha_{\mathcal{S}/\rho} = \begin{cases} \{(\mathrm{in}_i \, \alpha)\, \rho\} & \text{if } \alpha(\mathrm{in}_i) \in S, \\ \mathrm{in}_i \, \alpha & \text{if } \alpha(\mathrm{in}_i) \in \mathbf{O}_p, \\ \emptyset & \text{if } \mathrm{in}_i \, \alpha = \emptyset, \end{cases} $$

$$ (s\rho)\, t_{\mathcal{S}/\rho} = \begin{cases} \{(s\, t_{\mathcal{S}})\, \rho\} & \text{if } t_{\mathcal{S}}(s) \in S, \\ s\, t_{\mathcal{S}} & \text{if } t_{\mathcal{S}}(s) \in \mathbf{O}_p, \\ \emptyset & \text{if } s\, t_{\mathcal{S}} = \emptyset, \end{cases} $$

for all $i \in [n]$, $s \in S$, and $t \in [\omega]$.

Congruences and homomorphisms of flowchart schemes behave just like congruences and homomorphisms of algebras. For example, if φ is a homomorphism from a Σ-scheme $\mathcal{S}$ to a Σ-scheme $\mathcal{S}'$ then $\ker_\varphi$ is a congruence on $\mathcal{S}$, and there exists a surjective homomorphism $\varphi_1 : \mathcal{S} \to \mathcal{S}/\ker_\varphi$ and an injective homomorphism $\varphi_2 : \mathcal{S}/\ker_\varphi \to \mathcal{S}'$ such that $\varphi = \varphi_1 \circ \varphi_2$. Conversely, if ρ is a congruence on a scheme $\mathcal{S}$, the function mapping each state s to the congruence class $s\rho$ is a surjective homomorphism, the *natural homomorphism* from $\mathcal{S}$ to $\mathcal{S}/\rho$.

In the next definition we adopt the universal algebraic concept of a subalgebra to flowchart schemes.

Definition 4.1.3. Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : n \to p$ are Σ-schemes. $\mathcal{S}'$ is a *sub-scheme* of $\mathcal{S}$ if $S' \subseteq S$ and the identity function $\mathrm{id}_{S'}$ is a homomorphism from $\mathcal{S}'$ to $\mathcal{S}$. We call $\mathcal{S}'$ a *proper sub-scheme* of $\mathcal{S}$ if it is a sub-scheme of $\mathcal{S}$ and $S' \subset S$.

Each sub-scheme of a scheme $\mathcal{S}$ is totally determined by (and is usually identified with) its set of states. Note that when φ is a simulation from $\mathcal{S}$ to $\mathcal{S}'$, $\mathrm{Dom}(\varphi)$ is a sub-scheme of $\mathcal{S}$ and $\mathrm{Rng}(\varphi)$ is a sub-scheme of $\mathcal{S}'$.

Definition 4.1.4. Suppose that $\mathcal{S} : n \to p$ is a Σ-scheme. A state $s \in S$ is called *accessible* if $s = u_{\mathcal{S}}(\mathrm{in}_i)$ for some $i \in [n]$ and $u \in [\omega]^*$, i.e., when in the graphical representation, s lies on a directed walk from an input node. Moreover, s is called *strongly accessible* if $s = \alpha_{\mathcal{S}}(\mathrm{in}_i)$ for some $i \in [n]$, i.e., when s is the target of an edge from some input node. We call $\mathcal{S}$ a *(strongly) accessible scheme* if each of its states is (strongly) accessible.

We denote the set of all accessible states of $\mathcal{S}$ by $\mathrm{Acc}(\mathcal{S})$. It is not hard to see that $\mathrm{Acc}(\mathcal{S})$ is the smallest sub-scheme of $\mathcal{S}$, called the *accessible part* of $\mathcal{S}$. Therefore, a scheme is accessible if and only if it has no proper sub-schemes.

Lemma 4.1.4. *Suppose that $\mathcal{S}$ and $\mathcal{S}'$ are strongly equivalent Σ-schemes. Then* $\mathrm{Dom}({}_{\mathcal{S}}\Gamma_{\mathcal{S}'}) = \mathrm{Acc}(\mathcal{S})$ *and* $\mathrm{Rng}({}_{\mathcal{S}}\Gamma_{\mathcal{S}'}) = \mathrm{Acc}(\mathcal{S}')$. *Moreover,*

$$ {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} = {}_{\mathrm{Acc}(\mathcal{S})}\Gamma_{\mathcal{S}'} = {}_{\mathcal{S}}\Gamma_{\mathrm{Acc}(\mathcal{S}')} = {}_{\mathrm{Acc}(\mathcal{S})}\Gamma_{\mathrm{Acc}(\mathcal{S}')}. \quad \square $$

Lemma 4.1.5. *Suppose that $\mathcal{S}$ and $\mathcal{S}'$ are Σ-schemes and $\varphi : \mathcal{S} \to \mathcal{S}'$ is a homomorphism. Define $\psi := \varphi \cap (\mathrm{Acc}(\mathcal{S}) \times S')$, so that ψ is the restriction of φ to the accessible states of $\mathcal{S}$. Then $\psi = {}_{\mathcal{S}}\Gamma_{\mathcal{S}'}$ and ψ is a surjective homomorphism* $\mathrm{Acc}(\mathcal{S}) \to \mathrm{Acc}(\mathcal{S}')$.

Proof. ψ is clearly a homomorphism from $\mathrm{Acc}(\mathcal{S})$ to $\mathcal{S}'$ and, by Lemma 4.1.4, ${}_{\mathcal{S}}\Gamma_{\mathcal{S}'} = {}_{\mathrm{Acc}(\mathcal{S})}\Gamma_{\mathcal{S}'} \subseteq \psi$. Since $\mathrm{Dom}({}_{\mathcal{S}}\Gamma_{\mathcal{S}'}) = \mathrm{Dom}(\psi) = \mathrm{Acc}(\mathcal{S})$ and ψ is a function, it follows that $\psi = {}_{\mathcal{S}}\Gamma_{\mathcal{S}'}$ and $\mathrm{Rng}(\psi) = \mathrm{Rng}({}_{\mathcal{S}}\Gamma_{\mathcal{S}'}) = \mathrm{Acc}(\mathcal{S}')$. $\square$

The next lemma gives various (well known) characterizations of the strong equivalence relation $\approx$ of flowchart schemes. See [15], for example.

Lemma 4.1.6. *Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : n \to p$ are Σ-schemes. Then the following statements are equivalent:*

1. *$\mathcal{S}$ and $\mathcal{S}'$ are strongly equivalent.*
2. $\forall i \in [n] \;\; \forall u \in [\omega]^* \;\; (\mathrm{label}_S \cup \mathrm{id}_O)(u_{\mathcal{S}}(\mathrm{in}_i)) = (\mathrm{label}_{S'} \cup \mathrm{id}_O)(u_{\mathcal{S}'}(\mathrm{in}_i))$.
3. *The relation*

$$ \{(s, s') \in S \times S' \mid \exists i \in [n] \;\; \exists u \in [\omega]^* \;\; s = u_{\mathcal{S}}(\mathrm{in}_i) \wedge s' = u_{\mathcal{S}'}(\mathrm{in}_i)\} $$

   is a simulation from $\mathcal{S}$ to $\mathcal{S}'$.

4. *The two schemes* $\mathrm{Acc}(\mathcal{S})/\Theta_{\mathrm{Acc}(\mathcal{S})}$ *and* $\mathrm{Acc}(\mathcal{S}')/\Theta_{\mathrm{Acc}(\mathcal{S}')}$ *are isomorphic.* $\square$

Every simulation relation γ from a scheme $\mathcal{S}$ to a scheme $\mathcal{S}'$ determines a scheme whose states are the ordered pairs in γ.

Definition 4.1.5. Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : n \to p$ are strongly equivalent Σ-schemes and γ is a simulation from $\mathcal{S}$ to $\mathcal{S}'$. Then γ determines a Σ-scheme $[\gamma] = (\gamma, \alpha_{[\gamma]}, \delta_{[\gamma]}) : n \to p$, where

$$\mathrm{in}_i\, \alpha_{[\gamma]} = \begin{cases} \{(\alpha_{\mathcal{S}}(\mathrm{in}_i), \alpha_{\mathcal{S}'}(\mathrm{in}_i))\} & \text{if } \alpha_{\mathcal{S}}(\mathrm{in}_i) \in S, \\ \mathrm{in}_i\, \alpha_{\mathcal{S}} & \text{if } \alpha_{\mathcal{S}}(\mathrm{in}_i) \in \mathbf{O}_p, \\ \emptyset & \text{if } \mathrm{in}_i\, \alpha_{\mathcal{S}} = \emptyset, \end{cases}$$

$$(s, s')\, t_{[\gamma]} = \begin{cases} \{(t_{\mathcal{S}}(s), t_{\mathcal{S}'}(s'))\} & \text{if } t_{\mathcal{S}}(s) \in S, \\ s\, t_{\mathcal{S}} & \text{if } t_{\mathcal{S}}(s) \in \mathbf{O}_p, \\ \emptyset & \text{if } s\, t_{\mathcal{S}} = \emptyset, \end{cases}$$

for all $i \in [n]$, $(s, s') \in \gamma$, and $t \in [\omega]$. We call the schemes $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$ and $[{}_{\mathcal{S}}\Omega_{\mathcal{S}'}]$ the *minimal and maximal direct product* of $\mathcal{S}$ and $\mathcal{S}'$, respectively.

Thus, for each $i \in [n]$, the ith input node of the scheme $[\gamma]$ has an out-edge if and only if the ith input node of $\mathcal{S}$, and hence of $\mathcal{S}'$, has an out-edge. Moreover, when exists, the target of this out-edge is the ordered pair (s, s'), where s and s' are the targets of the out-edges of the ith input nodes of $\mathcal{S}$ and $\mathcal{S}'$, respectively. However, if say s is an output node, the target of the out-edge of the ith input node of $[\gamma]$ is the corresponding output node. Let $(s, s') \in \gamma$ and $t \in [\omega]$. Then, in the graphical representation of the scheme $[\gamma]$, the node (s, s') has an out-edge labeled by t if and only if s, and hence s' has an out-edge labeled by t. Suppose that v and v' denote the targets of these edges. Then, since γ is a simulation, v is an internal node if and only if v' is. In this case, the ordered pair $(v, v') \in \gamma$ is the target of the out-edge of (s, s') labeled by t. Otherwise s and s' are output nodes, and the target is the corresponding output node of $[\gamma]$.

It follows by Lemma 4.1.6 that two accessible schemes are strongly equivalent if and only if they have a common homomorphic image. This observation and the following two lemmas show that all pullbacks exist in the category in which the objects are the accessible Σ-schemes, and the morphisms are their homomorphisms.

Lemma 4.1.7. *Suppose that $\mathcal{S} : n \to p$ and $\mathcal{S}' : n \to p$ are strongly equivalent Σ-schemes. Then their minimal direct product $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$ is an accessible scheme. Moreover, the two projection functions $\pi : {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \to S$ and $\pi' : {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \to S'$ are homomorphisms, namely, $\pi = {}_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}\Gamma_{\mathcal{S}}$ and $\pi' = {}_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}\Gamma_{\mathcal{S}'}$.*

Proof. Suppose that $(s, s') \in {}_{\mathcal{S}}\Gamma_{\mathcal{S}'}$. By Lemma 4.1.3, there exists an integer $i \in [n]$ and a word $u \in [\omega]^*$ such that

$$(s, s') \;=\; (u_{\mathcal{S}}(\mathrm{in}_i), u_{\mathcal{S}'}(\mathrm{in}_i)) \;=\; u_{[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]}(\mathrm{in}_i),$$

showing (s, s') is an accessible state of $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$. It is trivial that the two projections are simulations, so they are homomorphisms. Now $\pi = {}_{[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]}\Gamma_{\mathcal{S}}$ and $\pi' = {}_{[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]}\Gamma_{\mathcal{S}'}$ by Lemma 4.1.5. □

Lemma 4.1.8. *Suppose that $\mathcal{S}$, $\mathcal{S}'$ and $\overline{\mathcal{S}}$ are Σ-schemes $n \to p$, $\varphi : \overline{\mathcal{S}} \to \mathcal{S}$ and $\varphi' : \overline{\mathcal{S}} \to \mathcal{S}'$ are homomorphisms. Then there exists a unique homomorphism ψ from $\mathrm{Acc}(\overline{\mathcal{S}})$ to $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$.*

Proof. By Lemma 4.1.5, the only possibility for ψ is the least simulation relation ${}_{\overline{\mathcal{S}}}\Gamma_{[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]}$, which is defined, since $\mathcal{S}$, $\mathcal{S}'$, $\overline{\mathcal{S}}$ and $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$ are strongly equivalent. To prove it is a function, assume that $u_{\overline{\mathcal{S}}}(\mathrm{in}_i) = v_{\overline{\mathcal{S}}}(\mathrm{in}_j)$ is a state of $\mathrm{Acc}(\overline{\mathcal{S}})$, for some integers $i, j \in [n]$ and words $u, v \in [\omega]^*$. Then

$$u_{\mathcal{S}}(\mathrm{in}_i) \;=\; \varphi(u_{\overline{\mathcal{S}}}(\mathrm{in}_i)) \;=\; \varphi(v_{\overline{\mathcal{S}}}(\mathrm{in}_j)) \;=\; v_{\mathcal{S}}(\mathrm{in}_j)$$

and

$$u_{\mathcal{S}'}(\mathrm{in}_i) \;=\; \varphi'(u_{\overline{\mathcal{S}}}(\mathrm{in}_i)) \;=\; \varphi'(v_{\overline{\mathcal{S}}}(\mathrm{in}_j)) \;=\; v_{\mathcal{S}'}(\mathrm{in}_j),$$

proving $u_{[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]}(\mathrm{in}_i) = v_{[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]}(\mathrm{in}_j)$. □

Lemma 4.1.9. *Suppose that $\mathcal{S} : n \to p$ is an accessible Σ-scheme and ρ is a congruence on $\mathcal{S}$. Then the minimal direct product $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}/\rho}]$ of $\mathcal{S}$ and $\mathcal{S}/\rho$ is isomorphic to $\mathcal{S}$.*

Proof. Since $\mathcal{S}$ is accessible, the states of $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}/\rho}]$ are all pairs $(s, s\,\rho)$, $s \in S$, and the projection $\pi : [{}_{\mathcal{S}}\Gamma_{\mathcal{S}/\rho}] \to \mathcal{S}$ is an isomorphism. □

4.1.1 Aperiodic congruences

In this subsection we define and study some special congruences of flowchart schemes, namely minimal, regular, simple and aperiodic congruences. Although the results of this subsection have little importance of their own, they serve as a technical basis in the course of proving our main result, the characterization of the Conway-equivalence of flowchart schemes.

Recall that when A and B are sets, we denote by $\mathrm{Const}[A, B]$ and $\mathrm{Biject}[A, B]$ the set of all constant functions and the set of all bijections $A \to B$, respectively. Suppose that ρ is a congruence on a scheme $\mathcal{S}$. The set of all *nonsingleton* equivalence classes of ρ will be denoted by $\mathbf{Cl}(\rho)$. Recall from Lemma 4.1.1 that the intersection of two (and in fact any nonzero number of) congruences on $\mathcal{S}$ is again a congruence on $\mathcal{S}$. It follows that if C' is a subset of an equivalence class C of a congruence ρ then there exists a least congruence $\Theta(C')$ on $\mathcal{S}$, called the *congruence generated by* C', such that $\Theta(C')$ identifies all the elements of C'. Note that $\Theta(C')$ is the least equivalence containing the relation

$$\Theta_0 = \{(\tau(a), \tau(b)) \mid a, b \in C',\ \tau \in [\omega]^*_{\mathcal{S}}[C, S]\} \;\subseteq\; S \times S$$

consisting of all pairs $(c,d) \in S \times S$ such that there exist $a, b \in C'$ and a word $u \in [\omega]^*$ such that c is the target of the directed walk from a labeled by u, and d is the target of the corresponding directed walk from b. The relation Θ_0 is usually not transitive, in which case $\Theta_0 \neq \Theta(C')$. Also note that if $|C'| \leq 1$, $\Theta(C')$ is the *trivial congruence* id_S on $\mathcal{S}$.

Definition 4.1.6. Suppose that $\mathcal{S} : n \to p$ is a Σ-scheme and ρ is a congruence on $\mathcal{S}$. The *rank* of ρ, denoted by $\#\rho$, is the cardinality of its largest congruence class. A congruence of rank k is also called a *k-congruence*. We say that ρ is

minimal if it is nontrivial and minimal among all nontrivial congruences of $\mathcal{S}$ with respect to set inclusion,

regular if it is generated by each one of its nonsingleton classes, i.e., if

$$\Theta(C) = \rho,$$

for all $C \in \mathbf{Cl}(\rho)$,

simple if

$$[\omega]^*_{\mathcal{S}}[C,D] \subseteq \mathrm{Const}[C,D] \cup \mathrm{Biject}[C,D],$$

for all $C, D \in \mathbf{Cl}(\rho)$,

aperiodic if

$$s \,\rho\, u_S(s) \Longrightarrow \exists k \geq 0 \;\; u^k_{\mathcal{S}}(s) = u^{k+1}_{\mathcal{S}}(s),$$

for all $s \in S$ and $u \in [\omega]^*$.

Note that a trivial congruence is simple, regular and aperiodic, by definition. Also note that every 2-congruence is simple and every minimal congruence is regular. However, there exist regular congruences which are not minimal. (For the simplest example, take the scheme $0 \to 0$ having three states labeled by a symbol σ_0 having no transitions. Then the relation that collapses all three states is a regular congruence which is clearly not minimal.)

The word "regular" is used here only as a technical term. The concept of regular congruence has nothing to do with regularity as used in automata theory. Nevertheless, the notion of aperiodic congruence stems from automata theory, since a congruence ρ is aperiodic if and only if for each congruence class C, the transformation semigroup $(C, [\omega]^*_{\mathcal{S}}[C,C])$, or the semigroup $[\omega]^*_{\mathcal{S}}[C,C]$ is aperiodic. See [24, 44].

Remark 4.1.1. Suppose that $\mathcal{S} : n \to p$ is a Σ-scheme and ρ is a congruence on $\mathcal{S}$. Then the following statements are equivalent.

1. ρ is aperiodic on $\mathcal{S}$.

2. None of the partial functions

$$\{u_{\mathcal{S}}[C', C'] \mid u \in [\omega]^*,\ C' \subseteq C \in \mathbf{Cl}(\rho)\}$$

is a nontrivial (cyclic) permutation.

3. $\exists k \in \mathbf{N}\ \forall C \in \mathbf{Cl}(\rho)\ \forall \tau \in [\omega]^*_{\mathcal{S}}[C, C]\ \ \tau^k = \tau^{k+1}$.

4. For all $C \in \mathbf{Cl}(\rho)$, no subsemigroup of the monoid $[\omega]^*_{\mathcal{S}}[C, C]$ is a nontrivial group.

In the next three lemmas we establish a few simple facts about the special congruences defined above.

Lemma 4.1.10. *Suppose that ρ is a simple congruence on the scheme $\mathcal{S}$. Then ρ is regular if and only if*

$$|[\omega]^*_{\mathcal{S}}[C, D] \cap \mathrm{Biject}[C, D]| \geq 1,$$

for all $C, D \in \mathbf{Cl}(\rho)$.

Proof. If ρ is simple and the above condition holds then ρ is clearly generated by any one of its nonsingleton equivalence classes. Now assume ρ is simple and the above condition fails, so that there are two nonsingleton equivalence classes C and D of ρ such that $[\omega]^*_{\mathcal{S}}[C, D] \cap \mathrm{Biject}[C, D] = \emptyset$. Then $[\omega]^*_{\mathcal{S}}[C, D] \subseteq \mathrm{Const}[C, D]$, and the congruence generated by the class C is properly contained in ρ, since it does not identify the elements of D. □

Lemma 4.1.11. *Suppose that ρ is a simple congruence on the scheme $\mathcal{S}$. Then ρ is aperiodic if and only if*

$$[\omega]^*_{\mathcal{S}}[C, C] \cap \mathrm{Biject}[C, C] = \{\mathrm{id}_C\},$$

for all $C \in \mathbf{Cl}(\rho)$.

Proof. Observe that the elements of $[\omega]^*_{\mathcal{S}}[C, C] \cap \mathrm{Biject}[C, C]$ form a subgroup in the monoid $[\omega]^*_{\mathcal{S}}[C, C]$. By Remark 4.1.1, this group has to be trivial. □

Lemma 4.1.12. *Suppose that ρ is a simple regular congruence on the scheme $\mathcal{S}$. Then ρ is aperiodic if and only if*

$$|[\omega]^*_{\mathcal{S}}[C, D] \cap \mathrm{Biject}[C, D]| = 1,$$

for all $C, D \in \mathbf{Cl}(\rho)$.

Proof. If ρ is a simple congruence satisfying the above condition then it is aperiodic by Lemma 4.1.11. Now assume ρ is simple, regular, and aperiodic. By Lemma 4.1.10 and Lemma 4.1.11, we only need to show that for all distinct nonsingleton equivalence classes C, D of ρ there exists at most one bijection in $[\omega]^*_{\mathcal{S}}[C, D]$. Suppose that τ and τ' are bijections in $[\omega]^*_{\mathcal{S}}[C, D]$. By Lemma 4.1.10, there exists a bijection $\pi \in [\omega]^*_{\mathcal{S}}[D, C]$. Now both functions $\tau \circ \pi$ and $\tau' \circ \pi$ are bijections in $[\omega]^*_{\mathcal{S}}[C, C]$, so they are equal, by Lemma 4.1.11. It follows that $\tau = \tau'$. □

Recall that when $\rho' \subseteq \rho$ are two equivalence relations on a set S, their quotient ρ/ρ', defined by

$$\forall s, s' \in S \quad (s/\rho')\ \rho/\rho'\ (s'/\rho') \iff s\ \rho\ s',$$

is an equivalence relation on the set S/ρ' of all equivalence classes of ρ'. It is not hard to see that when $\rho' \subseteq \rho$ are congruences on a scheme $\mathcal{S}$ then the equivalence ρ/ρ' is a congruence on the quotient scheme $\mathcal{S}/\rho'$ and $(\mathcal{S}/\rho')/(\rho/\rho')$ is isomorphic to $\mathcal{S}/\rho$. The following two lemmas show that some nice properties of ρ are inherited to ρ' and ρ/ρ'.

Lemma 4.1.13. *Suppose that ρ is an aperiodic congruence on the scheme $\mathcal{S}$. If $\rho' \subseteq \rho$ is a congruence on $\mathcal{S}$, then ρ' is aperiodic on $\mathcal{S}$, and the quotient congruence $\overline{\rho} = \rho/\rho'$ is aperiodic on the quotient scheme $\overline{\mathcal{S}} = \mathcal{S}/\rho'$.*

Proof. It is trivial that ρ' is aperiodic. Suppose that $C\ \overline{\rho}\ u_{\overline{\mathcal{S}}}(C)$ for some word $u \in [\omega]^*$ and congruence class $C = s/\rho'$. Then $s\ \rho\ u_{\mathcal{S}}(s)$, and since ρ is aperiodic, $u_{\mathcal{S}}^k(s) = u_{\mathcal{S}}^{k+1}(s) \in S$, for some integer $k \geq 0$. It follows that $u_{\overline{\mathcal{S}}}^k(C) = u_{\overline{\mathcal{S}}}^{k+1}(C)$. □

Lemma 4.1.14. *Suppose that ρ is a simple congruence on the scheme $\mathcal{S}$. If $\rho' \subseteq \rho$ is a congruence on $\mathcal{S}$ generated by a class $C \in \mathcal{S}/\rho$, then ρ' is simple on $\mathcal{S}$, and the quotient congruence $\overline{\rho} = \rho/\rho'$ is simple on the quotient scheme $\overline{\mathcal{S}} = \mathcal{S}/\rho'$. Moreover, if $C \in \mathbf{Cl}(\rho)$ then $|\mathbf{Cl}(\overline{\rho})| = |\mathbf{Cl}(\rho)| - |\mathbf{Cl}(\rho')| < |\mathbf{Cl}(\rho)|$.*

Proof. The case that $|C| = 1$ is trivial, so assume $C \in \mathbf{Cl}(\rho)$. Then

$$\mathbf{Cl}(\rho') \;=\; \{D \in \mathbf{Cl}(\rho) \mid [\omega]^*_{\mathcal{S}}[C, D] \cap \mathrm{Biject}[C, D] \neq \emptyset\} \;\subseteq\; \mathbf{Cl}(\rho).$$

Since ρ is simple, it follows that ρ' is simple. The nonsingleton equivalence classes of $\overline{\rho}$ are of the form

$$\overline{D} \;=\; \{\{d\} \mid d \in D\} \;=\; D/\mathrm{id}_D,$$

where D is a nonsingleton equivalence class of ρ which is not an equivalence class of ρ'. The map $D \mapsto \overline{D}$ is a bijection from $\mathbf{Cl}(\rho) \setminus \mathbf{Cl}(\rho')$ to $\mathbf{Cl}(\overline{\rho})$. In particular, since $\mathbf{Cl}(\rho') \subseteq \mathbf{Cl}(\rho)$, we have

$$|\mathbf{Cl}(\overline{\rho})| \;=\; |\mathbf{Cl}(\rho) \setminus \mathbf{Cl}(\rho')| \;=\; |\mathbf{Cl}(\rho)| - |\mathbf{Cl}(\rho')| \;<\; |\mathbf{Cl}(\rho)|.$$

Suppose that $\overline{D}$ and $\overline{E}$ are nonsingleton classes of $\overline{\rho}$. Then

$$u_{\overline{\mathcal{S}}}(\{d\}) = \{e\} \iff u_{\mathcal{S}}(d) = e,$$

for all $d \in D$, $e \in E$ and $u \in [\omega]^*$, showing that $u_{\overline{\mathcal{S}}}[\overline{D}, \overline{E}]$ is constant/bijective if and only if $u_{\mathcal{S}}[D, E]$ is constant/bijective. It follows that $\overline{\rho}$ is simple. □

Lemma 4.1.15. *Suppose that ρ is a simple congruence on the scheme $\mathcal{S}$. Then there exists a simple regular congruence $\rho' \subseteq \rho$ on $\mathcal{S}$ such that the congruence $\overline{\rho} = \rho/\rho'$ is simple on the quotient scheme $\overline{\mathcal{S}} = \mathcal{S}/\rho'$. Moreover, if ρ is aperiodic, so are ρ' and $\overline{\rho}$, and if ρ is nontrivial then ρ' is nontrivial and $|\mathbf{Cl}(\overline{\rho})| = |\mathbf{Cl}(\rho)| - |\mathbf{Cl}(\rho')| < |\mathbf{Cl}(\rho)|$.*

Proof. If ρ is trivial so is the claim, therefore assume that $|\mathbf{Cl}(\rho)| > 0$. Consider the congruences $\Theta(D)$ generated by the nonsingleton classes D of ρ. Since there are finitely many of them, there exists a minimal such congruence, i.e., there is a nonsingleton equivalence class C of ρ such that whenever $\Theta(D) \subseteq \Theta(C)$ for some $D \in \mathbf{Cl}(\rho)$, then $\Theta(D) = \Theta(C)$. Let ρ' be the congruence $\Theta(C)$. Then clearly $\rho' \subseteq \rho$, and both ρ' and $\overline{\rho}$ are simple by Lemma 4.1.14. If ρ is aperiodic then ρ' and $\overline{\rho}$ are also aperiodic by Lemma 4.1.13. It follows by Lemma 4.1.14 that $|\mathbf{Cl}(\overline{\rho})| = |\mathbf{Cl}(\rho)| - |\mathbf{Cl}(\rho')| < |\mathbf{Cl}(\rho)|$. To prove that ρ' is regular assume that D is a nonsingleton equivalence class of ρ'. Then $\Theta(D) \subseteq \rho'$ and, as noted in the proof of Lemma 4.1.14, D is a nonsingleton class of ρ. It follows by the minimality of ρ' that $\Theta(D) = \rho'$. □

Corollary 4.1.16. *Suppose that ρ is a simple aperiodic congruence on the scheme $\mathcal{S}$. Then there exists an integer $m \geq 1$, a sequence $\mathcal{S}_1, \ldots, \mathcal{S}_m$ of schemes, and a sequence $\rho_1, \ldots, \rho_{m-1}$ of simple, aperiodic and regular congruences such that*

$$\begin{aligned} \mathcal{S}_1 &= \mathcal{S}, \\ \mathcal{S}_m &= \mathcal{S}/\rho, \quad \textit{and} \\ \mathcal{S}_{i+1} &= \mathcal{S}_i/\rho_i, \end{aligned}$$

for all $i \in [m-1]$.

Proof. By a straightforward induction on $|\mathbf{Cl}(\rho)|$, using Lemma 4.1.15. □

Minimal congruences identify "as few states as possible", minimal 2-congruences are even more restricted. We end this subsection by showing that every simple aperiodic congruence can be "decomposed" into a sequence of minimal aperiodic 2-congruences.

Lemma 4.1.17. *Suppose that ρ is a nontrivial, simple, aperiodic and regular congruence on the scheme $\mathcal{S}$. Then there exists a minimal aperiodic 2-congruence $\rho' \subseteq \rho$ on $\mathcal{S}$ such that the quotient congruence $\overline{\rho} = \rho/\rho'$ is simple, aperiodic and regular on the quotient scheme $\overline{\mathcal{S}} = \mathcal{S}/\rho'$. Moreover, $\#\overline{\rho} = \#\rho - 1$.*

Proof. Let $C' = \{a, b\}$ be a two-element subset of a congruence class $C \in \mathbf{Cl}(\rho)$ and $\rho' := \Theta(C')$. Then clearly $\rho' \subseteq \rho$, and both ρ' and $\overline{\rho}$ are aperiodic by Lemma 4.1.13. We know from Lemma 4.1.12 that each set $[\omega]^*_{\mathcal{S}}[D, E]$ contains a unique bijection τ_{DE}, for all $D, E \in \mathbf{Cl}(\rho)$. It follows that $\tau_{DE} \circ \tau_{EF} = \tau_{DF}$ and $\tau_{DD} = \mathrm{id}_D$, for all $D, E, F \in \mathbf{Cl}(\rho)$. Now ρ' is the least equivalence relation containing

$$\begin{aligned} \beta &:= \{(\tau(a), \tau(b)) \mid \tau \in [\omega]^*_C[S, ,] \; \tau(a) \neq \tau(b)\} \\ &= \{(\tau_{CD}(a), \tau_{CD}(b)) \mid D \in \mathbf{Cl}(\rho)\}. \end{aligned}$$

Since β is a transitive relation, $\rho' = \beta \cup \beta^{-1} \cup \mathrm{id}_S$ is a 2-congruence. In order to prove that ρ' is minimal, assume that $\gamma \subseteq \rho'$ is a nontrivial congruence on $\mathcal{S}$. Then γ is generated by two states a', b' with $(a', b') \in \beta$, say $a' = \tau_{CD}(a)$ and $b' = \tau_{CD}(b)$, where D is a nonsingleton class of ρ. But then

$$a \;=\; \tau_{DC}(a') \;\gamma\; \tau_{DC}(b') \;=\; b,$$

and since ρ' is generated by $\{a, b\}$ it follows that $\gamma = \rho'$. The congruence classes of $\overline{\rho}$ are of the form

$$D/\rho' = \begin{cases} \{\{d\}\} & \text{if } D = \{d\}, \\ \{\{d\} \mid d \in D \setminus \{\tau_{CD}(a), \tau_{CD}(b)\}\} \cup \{\{\tau_{CD}(a), \tau_{CD}(b)\}\} & \text{if } |D| > 1, \end{cases}$$

where D is an equivalence class of ρ. This shows $\#\overline{\rho} = \#\rho - 1$, and also that $\overline{\rho}$ is simple and regular on $\overline{\mathcal{S}}$. □

Corollary 4.1.18. *Suppose that ρ is a simple aperiodic congruence on the scheme $\mathcal{S}$. Then there exists an integer $m \geq 1$, a sequence $\mathcal{S}_1, \ldots, \mathcal{S}_m$ of schemes, and a sequence $\rho_1, \ldots, \rho_{m-1}$ of minimal aperiodic 2-congruences such that*

$$\begin{aligned} \mathcal{S}_1 &= \mathcal{S}, \\ \mathcal{S}_m &= \mathcal{S}/\rho, \quad \textit{and} \\ \mathcal{S}_{i+1} &= \mathcal{S}_i/\rho_i, \end{aligned}$$

for all $i \in [m-1]$.

Proof. By induction on $\#\rho$, using Corollary 4.1.16 and Lemma 4.1.17. □

4.1.2 Aperiodic homomorphisms

This subsection is devoted to scheme homomorphisms having an aperiodic kernel, or aperiodic homomorphisms, for short. Based on these homomorphisms we define two relations $\to$ and $\Rightarrow$ on the Θ-algebra $\mathcal{S}ch_\Sigma$ of Σ-schemes, the first being strictly weaker than the second. Then we prove in Lemma 4.1.29 that the equivalences $\overset{*}{\leftrightarrow}$ and $\overset{*}{\Leftrightarrow}$ generated by these relations coincide, and that this equivalence is a congruence on $\mathcal{S}ch_\Sigma$. We also show that $\overset{*}{\Leftrightarrow}$ is just the composite of $\Leftarrow$ with $\Rightarrow$, where $\Leftarrow$ denotes the inverse of the relation $\Rightarrow$. This is done in two steps: in Lemma 4.1.23 we prove that the relation $\Leftarrow \circ \Rightarrow$ contains the relation $\Rightarrow \circ \Leftarrow$. In particular, it follows that $\overset{*}{\Leftrightarrow} = \overset{*}{\Leftarrow} \circ \overset{*}{\Rightarrow}$. Then in Lemma 4.1.25 we show that $\Rightarrow$ is reflexive and transitive, so that $\overset{*}{\Rightarrow} = \Rightarrow$, and $\overset{*}{\Leftarrow} = \Leftarrow$.

Definition 4.1.7. Suppose that $\mathcal{S}$ and $\mathcal{S}'$ are Σ-schemes and φ is a homomorphism from $\mathcal{S}$ to $\mathcal{S}'$. We write

$\mathcal{S} \overset{\varphi}{\to} \mathcal{S}'$ if φ is injective or $\ker_\varphi$ is a minimal aperiodic 2-congruence on $\mathcal{S}$,

$\mathcal{S} \overset{\varphi}{\Rightarrow} \mathcal{S}'$ if $\ker_\varphi$ is an aperiodic congruence on $\mathcal{S}$.

We define two relations on Σ-schemes by

$$\begin{aligned} \mathcal{S} \to \mathcal{S}' &\iff \exists \varphi \; \mathcal{S} \overset{\varphi}{\to} \mathcal{S}' \\ \mathcal{S} \Rightarrow \mathcal{S}' &\iff \exists \varphi \; \mathcal{S} \overset{\varphi}{\Rightarrow} \mathcal{S}'. \end{aligned}$$

The inverses of these relations are denoted by the corresponding reversed arrows, and we use the standard notation for the various closures. For example, $\stackrel{*}{\Rightarrow}$ is the least reflexive and transitive relation containing $\Rightarrow$, and $\stackrel{*}{\leftrightarrow}$ is the equivalence relation generated by $\rightarrow$.

Using these definitions we can rephrase Corollary 4.1.18 in the following form.

Corollary 4.1.19. *If ρ is a simple aperiodic congruence on a Σ-scheme $\mathcal{S}$ then $\mathcal{S} \stackrel{*}{\rightarrow} \mathcal{S}/\rho$.* □

We summarize the results of this subsection in the following proposition.

Proposition 4.1.20. *The equivalence relations $\stackrel{*}{\leftrightarrow}$ and $\stackrel{*}{\Leftrightarrow}$ agree on Sch_Σ. Moreover, $\stackrel{*}{\Leftrightarrow}$ is a congruence relation on Sch_Σ, and for all $\mathcal{S}, \mathcal{S}' : n \rightarrow p$ in Sch_Σ,*

$$\mathcal{S} \stackrel{*}{\Leftrightarrow} \mathcal{S}' \quad \textit{if and only if} \quad \mathcal{S} \stackrel{\pi}{\Leftarrow} [{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}] \stackrel{\pi'}{\Rightarrow} \mathcal{S}',$$

where $\pi : {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \rightarrow S$ and $\pi' : {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \rightarrow S'$ are the two projections.

First we prove that $\stackrel{*}{\Leftrightarrow}$ is a congruence on Sch_Σ by showing that the relation $\Rightarrow$ preserves the operations of pairing, composition and iteration. It then follows that $\Rightarrow$ also preserves the separated sum operation.

Lemma 4.1.21. *Suppose that $\mathcal{F}, \mathcal{G}, \mathcal{F}', \mathcal{G}'$ are Σ-schemes of appropriate sorts. Then*

$$\mathcal{F} \stackrel{\varphi}{\Rightarrow} \mathcal{F}' \ \wedge \ \mathcal{G} \stackrel{\psi}{\Rightarrow} \mathcal{G}' \Longrightarrow \langle \mathcal{F}, \mathcal{G} \rangle \stackrel{\varphi \cup \psi}{\Rightarrow} \langle \mathcal{F}', \mathcal{G}' \rangle$$

$$\mathcal{F} \stackrel{\varphi}{\Rightarrow} \mathcal{F}' \ \wedge \ \mathcal{G} \stackrel{\psi}{\Rightarrow} \mathcal{G}' \Longrightarrow \mathcal{F} \cdot \mathcal{G} \stackrel{\varphi \cup \psi}{\Rightarrow} \mathcal{F}' \cdot \mathcal{G}'$$

$$\mathcal{F} \stackrel{\varphi}{\Rightarrow} \mathcal{G} \Longrightarrow \mathcal{F}^\dagger \stackrel{\varphi}{\Rightarrow} \mathcal{G}^\dagger.$$

Proof. The first two implications can be handled in the same way, therefore we only prove the first one. Suppose that $\mathcal{F} \stackrel{\varphi}{\Rightarrow} \mathcal{F}'$ and $\mathcal{G} \stackrel{\psi}{\Rightarrow} \mathcal{G}'$. By Lemma 4.1.1, $\varphi \cup \psi$ is a simulation from $\langle \mathcal{F}, \mathcal{G} \rangle$ to $\langle \mathcal{F}', \mathcal{G}' \rangle$. Since the set of states of $\langle \mathcal{F}, \mathcal{G} \rangle$ is the disjoint union of those of $\mathcal{F}$ and $\mathcal{G}$, $\varphi \cup \psi$ is a function and $\ker_{\varphi \cup \psi} = \ker_\varphi \cup \ker_\psi$. If C is a congruence class of $\ker_{\varphi \cup \psi}$ then C is either a congruence class of $\ker_\varphi$ and $[\omega]^*_{\langle \mathcal{F}, \mathcal{G} \rangle}[C, C] = [\omega]^*_{\mathcal{F}}[C, C]$, or C is a congruence class of $\ker_\psi$ and $[\omega]^*_{\langle \mathcal{F}, \mathcal{G} \rangle}[C, C] = [\omega]^*_{\mathcal{G}}[C, C]$. Since $\ker_\varphi$ is aperiodic on $\mathcal{F}$ and $\ker_\psi$ is aperiodic on $\mathcal{G}$, $\ker_{\varphi \cup \psi}$ is aperiodic on $\langle \mathcal{F}, \mathcal{G} \rangle$.

As for the last implication, if $\mathcal{F} \stackrel{\varphi}{\Rightarrow} \mathcal{G}$ then φ is a homomorphism from $\mathcal{F}^\dagger$ to $\mathcal{G}^\dagger$, by Lemma 4.1.1. Suppose that C is a congruence class of $\ker_\varphi$. Looking at the definition of the iteration operation it is not hard to see that $[\omega]^*_{\mathcal{F}^\dagger}[C, C] \subseteq [\omega]^*_{\mathcal{F}}[C, C] \cup \mathrm{Const}[C, C]$. By Remark 4.1.1, $\ker_\varphi$ is aperiodic on $\mathcal{F}^\dagger$. □

Corollary 4.1.22. *$\stackrel{*}{\Leftrightarrow}$ is a congruence relation on Sch_Σ.* □

Next we give a simple characterization of the congruence $\stackrel{*}{\Leftrightarrow}$. After proving two lemmas, the results are summarized in Theorem 4.1.27.

Lemma 4.1.23. $(\Rightarrow \circ \Leftarrow) \subseteq (\Leftarrow \circ \Rightarrow)$.

Proof. Suppose that $\mathcal{S} \overset{\varphi}{\Rightarrow} \underline{\mathcal{S}} \overset{\varphi'}{\Leftarrow} \mathcal{S}'$ for some Σ-schemes $\mathcal{S}, \mathcal{S}', \underline{\mathcal{S}} : n \to p$. Then $\mathcal{S}$ and $\mathcal{S}'$ are strongly equivalent, so their minimal direct product $[\mathcal{S}\Gamma_{\mathcal{S}'}]$ exists. By Lemma 4.1.7, the two projection functions $\pi : {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \to S$ and $\pi' : {}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \to S'$ are homomorphisms from $[\mathcal{S}\Gamma_{\mathcal{S}'}]$ to $\mathcal{S}$ and $\mathcal{S}'$, respectively. In order to prove that $\ker_\pi$ is aperiodic on $[\mathcal{S}\Gamma_{\mathcal{S}'}]$, assume that

$$(s, s') \ker_\pi w_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}((s, s')),$$

for a word $w \in [\omega]^*$ and state (s, s') of $[\mathcal{S}\Gamma_{\mathcal{S}'}]$. Let us write (r, r') for $w_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}((s, s'))$, so that $r = w_{\mathcal{S}}(s)$ and $r' = w_{\mathcal{S}'}(s')$. Since $[\mathcal{S}\Gamma_{\mathcal{S}'}]$ is an accessible scheme, there exists an integer $i \in [n]$ and a word $u \in [\omega]^*$ such that

$$(s, s') \;=\; u_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}(\mathrm{in}_i) \;=\; (u_{\mathcal{S}}(\mathrm{in}_i), u_{\mathcal{S}'}(\mathrm{in}_i)).$$

Thus

$$\begin{aligned} s &= u_{\mathcal{S}}(\mathrm{in}_i) \\ s' &= u_{\mathcal{S}'}(\mathrm{in}_i) \\ r = w_{\mathcal{S}}(s) &= (uw)_{\mathcal{S}}(\mathrm{in}_i) \\ r' = w_{\mathcal{S}'}(s') &= (uw)_{\mathcal{S}'}(\mathrm{in}_i). \end{aligned}$$

By Lemma 4.1.5, there is a unique homomorphism from $[\mathcal{S}\Gamma_{\mathcal{S}'}]$ to $\underline{\mathcal{S}}$, and it follows by Lemma 4.1.1 that both functions $\pi \circ \varphi$ and $\pi' \circ \varphi'$ are homomorphisms from $[\mathcal{S}\Gamma_{\mathcal{S}'}]$ to $\underline{\mathcal{S}}$, so they are equal. It follows that

$$\varphi(s) \;=\; \varphi'(s')$$

and

$$\varphi(r) \;=\; \varphi'(r').$$

Since $(s, s') \ker_\pi (r, r')$ we have $s = r$ and

$$\varphi'(s') \;=\; \varphi(s) \;=\; \varphi(r) \;=\; \varphi'(r'),$$

so that $s' \ker_{\varphi'} r' = w_{\mathcal{S}'}(s')$. Since $\ker_{\varphi'}$ is aperiodic on $\mathcal{S}'$, there exists an integer $k \geq 0$ such that

$$w^k_{\mathcal{S}'}(s') = w^{k+1}_{\mathcal{S}'}(s').$$

On the other hand, since $s = r = w_{\mathcal{S}}(s)$,

$$w^k_{\mathcal{S}}(s) = w^{k+1}_{\mathcal{S}}(s).$$

It follows that

$$w^k_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}((s, s')) = w^{k+1}_{[\mathcal{S}\Gamma_{\mathcal{S}'}]}((s, s')),$$

proving $\ker_\pi$ is aperiodic on $[\mathcal{S}\Gamma_{\mathcal{S}'}]$. A similar argument shows that $\ker_{\pi'}$ is aperiodic. □

Corollary 4.1.24. $\overset{*}{\Leftrightarrow} = (\overset{*}{\Leftarrow} \circ \overset{*}{\Rightarrow})$. □

Lemma 4.1.25. $\Rightarrow = \overset{*}{\Rightarrow}$

Proof. We have to show that $\Rightarrow$ is reflexive and transitive. Since each trivial congruence is aperiodic, $\Rightarrow$ is reflexive. To prove it is transitive, assume that $\mathcal{F} \overset{\varphi}{\Rightarrow} \mathcal{G} \overset{\psi}{\Rightarrow} \mathcal{H}$. Then the composite function $\varphi \circ \psi$ is a homomorphism from $\mathcal{F}$ to $\mathcal{H}$, and the result follows if we can prove that $\ker_{\varphi\circ\psi}$ is aperiodic on $\mathcal{F}$. Suppose that $s \ker_{\varphi\circ\psi} u_{\mathcal{F}}(s)$ for some word $u \in [\omega]^*$ and state s of $\mathcal{F}$. Then

$$\varphi(s) \ker_{\psi} \varphi(u_{\mathcal{F}}(s)) = u_{\mathcal{G}}(\varphi(s)).$$

Since $\ker_{\psi}$ is aperiodic, there exists an integer $k \geq 0$ such that

$$u_{\mathcal{G}}^{k}(\varphi(s)) = u_{\mathcal{G}}^{k+1}(\varphi(s)).$$

Let us write s' for $u_{\mathcal{F}}^{k}(s)$. Then

$$\begin{aligned}\varphi(s') &= u_{\mathcal{G}}^{k}(\varphi(s)) \\ &= u_{\mathcal{G}}^{k+1}(\varphi(s)) \\ &= u_{\mathcal{G}}(u_{\mathcal{G}}^{k}(\varphi(s))) \\ &= u_{\mathcal{G}}(\varphi(s')) \\ &= \varphi(u_{\mathcal{F}}(s')).\end{aligned}$$

Thus $s' \ker_{\varphi} u_{\mathcal{F}}(s')$, and since $\ker_{\varphi}$ is aperiodic on $\mathcal{F}$, there exists an integer $l \geq 0$ such that

$$u_{\mathcal{F}}^{k+l}(s) = u_{\mathcal{F}}^{l}(s') = u_{\mathcal{F}}^{l+1}(s') = u_{\mathcal{F}}^{(k+l)+1}(s).$$ □

□

Corollary 4.1.26. $\overset{*}{\Leftrightarrow} = (\Leftarrow \circ \Rightarrow)$. □

Theorem 4.1.27. *Suppose that $\mathcal{S}$ and $\mathcal{S}'$ are Σ-schemes $n \to p$. Then $\mathcal{S} \overset{*}{\Leftrightarrow} \mathcal{S}'$ if and only if $\mathcal{S}$ and $\mathcal{S}'$ are strongly equivalent and $\mathcal{S} \overset{\pi}{\Leftarrow} [_{\mathcal{S}}\Gamma_{\mathcal{S}'}] \overset{\pi'}{\Rightarrow} \mathcal{S}'$, where π and π' are the two projections.*

Proof. Trivially, the above condition is sufficient. To prove it is necessary assume that $\mathcal{S} \overset{*}{\Leftrightarrow} \mathcal{S}'$. Then $\mathcal{S}$ and $\mathcal{S}'$ are strongly equivalent schemes, therefore $[_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$ exists. By Corollary 4.1.26, there also exists a scheme $\overline{\mathcal{S}}$ such that $\mathcal{S} \overset{\varphi}{\Leftarrow} \overline{\mathcal{S}} \overset{\varphi'}{\Rightarrow} \mathcal{S}'$. Let γ and γ' be the restrictions of φ and φ' to the accessible states of $\overline{\mathcal{S}}$, respectively. Then $\gamma : \mathrm{Acc}(\overline{\mathcal{S}}) \to \mathcal{S}$ and $\gamma' : \mathrm{Acc}(\overline{\mathcal{S}}) \to \mathcal{S}'$, by Lemma 4.1.5. Further, $\ker_{\gamma}$ and $\ker_{\gamma'}$ are aperiodic congruences on $\mathrm{Acc}(\overline{\mathcal{S}})$, so we have $\mathcal{S} \overset{\gamma}{\Leftarrow} \mathrm{Acc}(\overline{\mathcal{S}}) \overset{\gamma'}{\Rightarrow} \mathcal{S}'$. Let ψ be the unique surjective homomorphism from $\mathrm{Acc}(\overline{\mathcal{S}})$ to $[_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$, which exists by Lemma 4.1.8. It follows by Lemma 4.1.5 that $\psi \circ \pi = \gamma = {}_{\overline{\mathcal{S}}}\Gamma_{\mathcal{S}}$ and $\psi \circ \pi' = \gamma' = {}_{\overline{\mathcal{S}}}\Gamma_{\mathcal{S}'}$. Lastly, Lemma 4.1.13 shows that the congruences $\ker_{\psi}$, $\ker_{\pi}$ and $\ker_{\pi'}$ are aperiodic, completing the proof. □

Corollary 4.1.28. *Suppose that $\mathcal{S}$ is an accessible Σ-scheme and ρ is a congruence on $\mathcal{S}$. Then $\mathcal{S} \overset{*}{\leftrightarrow} \mathcal{S}/\rho$ if and only if ρ is aperiodic.*

Proof. By Lemma 4.1.9 and Theorem 4.1.27. □

It is obvious that the relation $\Rightarrow$ properly contains the relation $\rightarrow$. We can even give examples when $\mathcal{S} \Rightarrow \mathcal{S}'$ holds, but $\mathcal{S} \overset{*}{\rightarrow} \mathcal{S}'$ does not. However, the next Lemma shows that $\Rightarrow$ is contained in the equivalence relation generated by $\rightarrow$. It follows then that the two equivalence relations $\overset{*}{\leftrightarrow}$ and $\overset{*}{\Leftrightarrow}$ are the same.

Lemma 4.1.29. $\Rightarrow \subseteq \overset{*}{\leftrightarrow}$.

Proof. Suppose that $\mathcal{S}$ and $\mathcal{F}$ are Σ-schemes $n \rightarrow p$ with $\mathcal{S} \Rightarrow \mathcal{F}$. Then there exists a homomorphism $\varphi : \mathcal{S} \rightarrow \mathcal{F}$ such that $\ker_\varphi$ is an aperiodic congruence on $\mathcal{S}$. As noted before, every homomorphism admits a surjective-injective factorization, i.e., there exists a surjective homomorphism $\varphi_1 : \mathcal{S} \rightarrow \mathcal{S}/\ker_\varphi$, and an injective homomorphism $\varphi_2 : \mathcal{S}/\ker_\varphi \rightarrow \mathcal{F}$ such that $\varphi = \varphi_1 \circ \varphi_2$. Let us denote $\ker_\varphi$ by ρ. Then $\mathcal{S}/\rho \overset{\varphi_2}{\rightarrow} \mathcal{F}$, and the result follows if we show that $\mathcal{S} \overset{*}{\leftrightarrow} \mathcal{S}/\rho$. To prove this we use induction on $\#\rho$. The base case $\#\rho = 1$ is trivial, so assume for the induction step that $\#\rho > 1$.

First we modify the input and transition functions of $\mathcal{S}$ to obtain a new scheme $\mathcal{S}' = (S, \alpha_{\mathcal{S}'}, \delta_{\mathcal{S}'}) : n \rightarrow p$. The difference between $\delta_\mathcal{S}$ and $\delta_{\mathcal{S}'}$ is that if C, D are congruence classes of ρ with $|D| = \#\rho$ and t is an integer such that $t_\mathcal{S}[C, D]$ is a non-surjective function $C \rightarrow D$, then we select an arbitrary element $d \in D \setminus \mathrm{Rng}(t_\mathcal{S}[C, D])$ and define $t_{\mathcal{S}'}[C, D]$ to be the constant function with value d. Similarly, for all $i \in [n]$, if $\alpha_\mathcal{S}(\mathrm{in}_i) \in S$ and the congruence class $D = \alpha_\mathcal{S}(\mathrm{in}_i)\,\rho$ has exactly $\#\rho$ elements, then we select an arbitrary element $d \in D \setminus \{\alpha_\mathcal{S}(\mathrm{in}_i)\}$ and define $\alpha_{\mathcal{S}'}(\mathrm{in}_i) := d$.

Note that for all words $u \in [\omega]^*$ and congruence classes C, D of ρ, either $u_{\mathcal{S}'}[C, D] = u_\mathcal{S}[C, D]$ or $u_\mathcal{S}[C, D]$ is a non-surjective function $C \rightarrow D$ and $u_{\mathcal{S}'}[C, D]$ is a constant function $C \rightarrow D$ such that $\mathrm{Rng}(u_{\mathcal{S}'}[C, D]) \cap \mathrm{Rng}(u_\mathcal{S}[C, D]) = \emptyset$. It follows that ρ is also an aperiodic congruence on $\mathcal{S}'$ and

$$\mathcal{S} \overset{\varphi_1}{\Rightarrow} \mathcal{S}/\rho \;=\; \mathcal{S}'/\rho \overset{\varphi_1}{\Leftarrow} \mathcal{S}'.$$

Thus $\mathcal{S} \overset{\pi}{\Leftarrow} [_\mathcal{S}\Gamma_{\mathcal{S}'}] \overset{\pi'}{\Rightarrow} \mathcal{S}'$, by Theorem 4.1.27.

Next we prove that

$$\forall i \in [n]\ \ \forall u \in [\omega]^*\ \ (u_\mathcal{S}(\mathrm{in}_i) \in S\ \wedge\ |u_\mathcal{S}(\mathrm{in}_i)\,\rho| = \#\rho) \Longrightarrow u_\mathcal{S}(\mathrm{in}_i) \neq u_{\mathcal{S}'}(\mathrm{in}_i). \quad (4.3)$$

The proof is by induction on the length of u. If u is the empty word ϵ and the congruence class $u_\mathcal{S}(\mathrm{in}_i)\,\rho$ has exactly $\#\rho$ elements then

$$u_\mathcal{S}(\mathrm{in}_i) \;=\; \alpha_\mathcal{S}(\mathrm{in}_i) \;\neq\; \alpha_\mathcal{S}(\mathrm{in}_i) \;=\; u_{\mathcal{S}'}(\mathrm{in}_i),$$

by the definition of $\alpha_{\mathcal{S}'}$. For the induction step, assume that $u = vt$, where $v \in [\omega]^*$ and $t \in [\omega]$. Let $c := v_\mathcal{S}(\mathrm{in}_i)$, $c' := v_{\mathcal{S}'}(\mathrm{in}_i)$, $d := u_\mathcal{S}(\mathrm{in}_i) = t_\mathcal{S}(c)$, $d' := u_{\mathcal{S}'}(\mathrm{in}_i) =$

$t_{\mathcal{S}'}(c')$, $C := c\rho = c'\rho$, $D := d\rho = d'\rho$. Suppose moreover that $|D| = \#\rho$. If the function $t_{\mathcal{S}}[C, D]$ is not surjective then $d' \notin \mathrm{Rng}(t_{\mathcal{S}}[C, D])$, by the definition of $\delta_{\mathcal{S}'}$. Since $d = t_{\mathcal{S}}(c) \in \mathrm{Rng}(t_{\mathcal{S}}[C, D])$, $d \neq d'$. If $t_{\mathcal{S}}[C, D]$ is surjective then $t_{\mathcal{S}}[C, D] = t_{\mathcal{S}'}[C, D]$ is necessarily a bijection, since $\#\rho = |D| \leq |C| \leq \#\rho$. Using the induction hypothesis we get $c \neq c'$, and thus

$$d = t_{\mathcal{S}}(c) \neq t_{\mathcal{S}}(c') = t_{\mathcal{S}'}(c') = d'.$$

Returning to the main proof, observe that ρ is not just a congruence on both schemes $\mathcal{S}$ and $\mathcal{S}'$, but it is also a simulation from $\mathcal{S}$ to $\mathcal{S}'$. Since ${}_{\mathcal{S}}\Gamma_{\mathcal{S}'}$ is the least simulation, ${}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \subseteq \rho$. Moreover, it follows from (4.3) that

$${}_{\mathcal{S}}\Gamma_{\mathcal{S}'} \subseteq \rho \setminus \{(s, s) \mid s \in S,\ |s\rho| = \#\rho\}.$$

Therefore, if (s, s') is a state of $[{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$ then

$$(s, s')\ker_\pi \subseteq \begin{cases} \{(s, x) \mid x \in s/\rho\} & \text{if } |s\rho| < \#\rho, \\ \{(s, x) \mid x \in s/\rho,\ x \neq s\} & \text{if } |s\rho| = \#\rho, \end{cases}$$

showing that $\#\ker_\pi < \#\rho$. By the same argument, $\#\ker_{\pi'} < \#\rho$. Thus, using the induction hypothesis, $\mathcal{S} \overset{*}{\leftrightarrow} [{}_{\mathcal{S}}\Gamma_{\mathcal{S}'}] \overset{*}{\leftrightarrow} \mathcal{S}'$.

Let ρ' denote the equivalence relation on S whose nonsingleton equivalence classes are those equivalence classes C of ρ with $1 < |C| < \#\rho$, i.e.,

$$s\,\rho'\,s' \iff s = s' \ \vee\ (s\,\rho\,s' \ \wedge\ |s\rho| < \#\rho),$$

for all $s, s' \in S$. Then ρ' is not necessarily a congruence on the scheme $\mathcal{S}$, but it *is* a congruence on $\mathcal{S}'$. This follows from the fact that for all congruence classes C, D of ρ with $|C| < |D| = \#\rho$, $[\omega]^*_{\mathcal{S}'}[C, D] \subseteq \mathrm{Const}[C, D]$. Moreover, $\rho' \subseteq \rho$ and $\#\rho' < \#\rho$. Let $\overline{\mathcal{S}}$ denote the quotient scheme $\mathcal{S}'/\rho'$, and let $\overline{\rho}$ be the quotient congruence ρ/ρ' on $\overline{\mathcal{S}}$. Then, by Lemma 4.1.13, ρ' is aperiodic on $\mathcal{S}'$ and $\overline{\rho}$ is aperiodic on $\overline{\mathcal{S}}$. We can apply the induction hypothesis once again to obtain $\mathcal{S}' \overset{*}{\leftrightarrow} \overline{\mathcal{S}}$.

Lastly, each nonsingleton congruence class of $\overline{\rho}$ has exactly $\#\rho$ elements, and, by the definition of $\mathcal{S}'$, $\overline{\rho}$ is a simple congruence on $\overline{\mathcal{S}}$. It follows by Corollary 4.1.19 that $\overline{\mathcal{S}} \overset{*}{\to} \overline{\mathcal{S}}/\overline{\rho} = \mathcal{S}'/\rho = \mathcal{S}/\rho$, completing the proof. □

Corollary 4.1.30. $\overset{*}{\leftrightarrow} = \overset{*}{\leftrightarrows}$. □

4.2 The free iteration theories

Although our interest is in the free Conway theories, we briefly review the description of the free iteration theories. All results in this section are well known and can be found in the book [15].

Suppose that Σ is a signature, and recall from Corollary 4.1.2 that the strong equivalence relation $\approx$ is a congruence on $\mathcal{S}ch_\Sigma$.

Proposition 4.2.1. *The quotient algebra $Sch_\Sigma/\approx$ is freely generated by Σ in the variety of all iteration theories.* □

Another description of the free Σ-generated iteration theory uses regular Σ-trees, cf. [15]. For a detailed study of infinite and regular trees see also [22].

Proposition 4.2.2. *It can be decided in polynomial time whether two Σ-schemes are strongly equivalent. Consequently, there exists a polynomial time algorithm which decides if an annotated Θ-equation $t = t'$ holds in all iteration theories.* □

4.3 The free Conway theories

In this section we finally complete the characterization of the free Conway theories.

Recall that in Definition 3.3.1, we defined $\equiv$ to be the least congruence on the Θ-algebra Sch_Σ of all Σ-schemes such that the quotient $Sch_\Sigma/\equiv$ satisfies the meta-equations (3.33) and (3.34). We proved in Theorem 3.3.2 that $Sch_\Sigma/\equiv$ is the free Conway theory generated by the signature Σ. Then we defined two more congruences $\overset{*}{\leftrightarrow}$ and $\overset{*}{\Leftrightarrow}$ using aperiodic simulations of flowchart schemes and proved that they are equal. A characterization of $\overset{*}{\Leftrightarrow}$ was given in Theorem 4.1.27.

Lemma 4.3.1. $\equiv \; = \; \overset{*}{\leftrightarrow}$.

Proof. In order to prove the containment $\equiv \subseteq \overset{*}{\leftrightarrow}$ we need to show that $Sch_\Sigma/\overset{*}{\leftrightarrow}$ satisfies the meta-equations (3.33) and (3.34). Suppose that $\mathcal{F} : n + m \to p$ is a Σ-scheme and let $\mathcal{G} := \langle \kappa_{n,m} \cdot \mathcal{F}, \lambda_{n,m} \cdot \mathcal{F} \rangle$. Then each state s of $\mathcal{F}$ has two copies in $\mathcal{G}$. Let $\varphi : G \to F$ be the function mapping both copies of s to s. Then φ is a surjective homomorphism, and $\ker_\varphi$ is a simple aperiodic 2-congruence on $\mathcal{G}$. In fact, if C and D are two congruence classes of $\ker_\varphi$ then $|C| = |D| = 2$, and $[\omega]^*_{\mathcal{G}}[C, D] \subseteq \mathrm{Biject}[C, D]$. Thus $\mathcal{G} \to \mathcal{G}/\ker_\varphi = \mathcal{F}$, showing that $Sch_\Sigma/\overset{*}{\leftrightarrow}$ satisfies the meta-equations (3.33).

In order to prove that $Sch_\Sigma/\overset{*}{\leftrightarrow}$ also satisfies (3.34), suppose that $\mathcal{F}$ is a Σ-scheme of sort $0 \to p$, for some $p \in \mathbf{N}$. Then the empty function is trivially an injective homomorphism from the base scheme 0_p to $\mathcal{F}$, so that $0_p \overset{*}{\leftrightarrow} \mathcal{F}$.

The reverse containment $\overset{*}{\leftrightarrow} \subseteq \equiv$ follows if we show that $\to \subseteq \equiv$. Suppose that $\mathcal{S} \overset{\varphi}{\to} \mathcal{S}'$ for Σ-schemes $\mathcal{S}, \mathcal{S}' : n \to p$ and a homomorphism $\varphi : \mathcal{S} \to \mathcal{S}'$. If φ is injective then

$$\mathcal{S} = \alpha \cdot \langle \mathcal{F}, \mathbf{1}_p \rangle$$

and

$$\mathcal{S}' = \alpha \cdot \langle (\mathbf{1}_r \oplus 0_m) \cdot \langle \mathcal{F} \cdot (0_{r+m} \oplus \mathbf{1}_p), \mathcal{G} \cdot (\mathbf{1}_r \oplus 0_m \oplus \mathbf{1}_p) \rangle^\dagger, \mathbf{1}_p \rangle,$$

for some Σ-schemes $\mathcal{F} : r \to p$, $\mathcal{G} : m \to r + p$ and partial base scheme $\alpha : n \to r + p$. Without going into the details we just note that $\mathcal{F}$ has the same states as $\mathcal{S}$, and the

states of $\mathcal{G}$ are those states of $\mathcal{S}'$ not in the range of φ. Moreover, r is the number of states in $\mathcal{F}$, m is the number of states in $\mathcal{G}$, and both $\mathcal{F}$ and $\mathcal{G}$ are strongly accessible. Since the equation

$$\mathcal{F} = (\mathbf{1}_r \oplus 0_m) \cdot \langle \mathcal{F} \cdot (0_{r+m} \oplus \mathbf{1}_p), \mathcal{G} \cdot (\mathbf{1}_r \oplus 0_m \oplus \mathbf{1}_p) \rangle^\dagger$$

holds in any Conway theory, it follows by Theorem 3.3.2 that $\mathcal{S} \equiv \mathcal{S}'$.

The second possibility is that $\ker_\varphi$ is a minimal aperiodic 2-congruence on $\mathcal{S}$. Then $\mathcal{S} \overset{\varphi_1}{\to} \mathcal{S}/\ker_\varphi \overset{\varphi_2}{\to} \mathcal{S}'$, where φ_1 is the natural homomorphism and φ_2 is injective. We have just proved above that $\mathcal{S}/\ker_\varphi \equiv \mathcal{S}'$. On the other hand,

$$\mathcal{S} = \alpha \cdot \langle \langle \langle \mathcal{F}, \mathcal{F} \rangle, \mathcal{G} \rangle^\dagger, \mathbf{1}_p \rangle$$

and

$$\mathcal{S}/\ker_\varphi = \alpha \cdot \langle \beta \cdot (\langle \mathcal{F}, \mathcal{G} \rangle \cdot (\beta \oplus \mathbf{1}_p))^\dagger, \mathbf{1}_p \rangle,$$

for some Σ-schemes $\mathcal{F} : r \to 2r+m+p$, $\mathcal{G} : m \to 2r+m+p$ and partial base scheme $\alpha : n \to 2r+m+p$, where β denotes the base scheme $\langle \mathbf{1}_r, \mathbf{1}_r \rangle \oplus \mathbf{1}_m : 2r+m \to r+m$. Now $\mathcal{S} \equiv \mathcal{S}/\ker_\varphi$ follows by Lemma 3.3.1. □

We have proved the following

Theorem 4.3.2. *$Sch_\Sigma/\overset{*}{\leftrightarrows}$ is freely generated by the signature Σ in the class of all Conway theories.*

Proof. By Theorem 3.3.2, Corollary 4.1.30 and Lemma 4.3.1. □

We know from Theorem 3.2.4 that, for an arbitrary $\mathbf{N} \times \mathbf{N}$-sorted set X, the Conway theory freely generated by X is isomorphic to the Conway theory freely generated by the signature $\Sigma(X)$. Thus, Theorem 4.3.2 describes all free Conway theories.

Let *Conway* denote the class of all Conway theories, that is, the class of all Θ-algebras satisfying the meta equations (3.20–3.34) and (3.46–3.48).

Recall that the equational theory of Conway theories is the collection $\mathbf{Eq}(\mathit{Conway}) = \mathbf{Eq}_\mathbf{X}(\mathit{Conway})$ of all annotated Θ-equations with variables in $\mathbf{X}$ which hold in all Conway theories, where $\mathbf{X}$ is some fixed $\mathbf{N} \times \mathbf{N}$-sorted set of variable symbols consisting of infinitely many elements of each sort $(n,p) \in \mathbf{N} \times \mathbf{N}$. Our last goal is to show that $\mathbf{Eq}(\mathit{Conway})$ is **PSPACE**-complete, i.e., it is **PSPACE**-complete to decide if an annotated Θ-equation with variables in $\mathbf{X}$ holds in all Conway theories.

We shall also consider the following decision problems:

- Aperiodic Σ-schemes ($\mathbf{ASch}_\Sigma$):

  INSTANCE: A (strongly accessible) Σ-scheme $\mathcal{S}$.

  QUESTION: Is $S \times S$ an aperiodic congruence on $\mathcal{S}$?

- Aperiodic congruences of Σ-schemes ($\mathbf{ACong}_\Sigma$):

  INSTANCE: A (strongly accessible) Σ-scheme $\mathcal{S}$ and a Σ-sorted relation $\rho \subseteq S \times S$.

  QUESTION: Is ρ an aperiodic congruence on $\mathcal{S}$?

- The Conway-equivalence problem of Σ-schemes ($\mathbf{SchEq}_\Sigma$):

  INSTANCE: A pair $(\mathcal{S}, \mathcal{S}')$ of Σ-schemes.

  QUESTION: Does $\mathcal{S} \equiv \mathcal{S}'$ hold?

- The equational theory of Conway theories with variables in Σ ($\mathbf{Eq}_\Sigma(\mathit{Conway})$):

  INSTANCE: A pair (t, t') of annotated Θ-terms with variables in Σ.

  QUESTION: Does the equation $t = t'$ hold in all Conway theories?

Recall the decision problem $\mathbf{ASF}_R$ from section 2.1, page 22.

Theorem 4.3.3. *Suppose that Σ is a signature containing at least one symbol σ_0 of rank $m \geq 2$. Then the decision problems $\mathbf{ASch}_\Sigma$, $\mathbf{ACong}_\Sigma$, $\mathbf{SchEq}_\Sigma$, $\mathbf{Eq}_\Sigma(\mathit{Conway})$, and $\mathbf{Eq}(\mathit{Conway})$ are **PSPACE**-complete with respect to logspace reductions.*

Proof. We shall prove that $\mathbf{ASF}_R \leq_{log} \mathbf{ASch}_\Sigma \leq_{log} \mathbf{ACong}_\Sigma \leq_{log} \mathbf{SchEq}_\Sigma \leq_{log} \mathbf{Eq}_\Sigma(\mathit{Conway}) \leq_{log} \mathbf{Eq}(\mathit{Conway}) \in \mathbf{PSPACE}$.

Proof of $\mathbf{ASF}_R \leq_{log} \mathbf{ASch}_\Sigma$. Suppose that $\mathcal{A}$ is an n-state deterministic automaton with input alphabet $\{0, 1\}$. We construct a strongly accessible Σ-scheme $\mathcal{S} : n \to 0$ such that $\mathcal{A}$ recognizes a star-free language (that is, $\mathcal{A}$ is aperiodic) if and only if $S \times S$ is an aperiodic congruence on $\mathcal{S}$. The states of $\mathcal{S}$ are the states of $\mathcal{A}$, each labeled by the symbol σ_0. The input function of $\mathcal{S}$ is an arbitrary bijection from $\mathbf{I}_n$ to S, and the transition function of $\mathcal{S}$ is defined by

$$t_\mathcal{S} = \begin{cases} 0_\mathcal{A} & \text{if } t = 1, \\ 1_\mathcal{A} & \text{if } 2 \leq t \leq m, \end{cases}$$

for all integers $t \in [m]$.

Proof of $\mathbf{ASch}_\Sigma \leq_{log} \mathbf{ACong}_\Sigma$. The map $\mathcal{S} \mapsto (\mathcal{S}, S \times S)$ is a logspace reduction.

Proof of $\mathbf{ACong}_\Sigma \leq_{log} \mathbf{SchEq}_\Sigma$. Suppose that $\mathcal{S} : n \to p$ is a strongly accessible Σ-scheme and $\rho \subseteq S \times S$ is a Σ-sorted relation. If ρ is not a congruence on $\mathcal{S}$ then $(\mathcal{S}, \rho)$ is mapped to some fixed pair $(\mathcal{F}, \mathcal{G})$ of Σ-schemes such that $\mathcal{F} \not\equiv \mathcal{G}$. Otherwise $(\mathcal{S}, \rho)$ is mapped to the pair $(\mathcal{S}, \mathcal{S}/\rho)$. All these calculations can be done in logarithmic space. The correctness of the reduction follows by Corollary 4.1.28.

Proof of $\mathbf{SchEq}_\Sigma \leq_{log} \mathbf{Eq}_\Sigma(\mathit{Conway})$. Let $\psi : \mathcal{T}_\Theta(\Sigma) \to \mathcal{S}ch_\Sigma$ denote the unique homomorphism mapping each symbol $\sigma \in \Sigma_q$ to the corresponding atomic scheme $\widehat{\sigma} : 1 \to q$. It is easy to find a logspace algorithm which, given a Σ-scheme $\mathcal{S} : n \to p$, constructs an annotated Θ-term $t_\mathcal{S} \in \mathcal{T}_\Theta(\Sigma)$ of sort (n, p) such that $\psi(t_\mathcal{S}) = \mathcal{S}$. The map $(\mathcal{S}, \mathcal{S}') \mapsto (t_\mathcal{S} = t'_\mathcal{S})$ is a logspace reduction.

Proof of $\mathbf{Eq}_\Sigma(Conway) \leq_{log} \mathbf{Eq}(Conway)$. This is trivial.

Proof of $\mathbf{Eq}(Conway) \in \mathbf{PSPACE}$. We outline a nondeterministic polynomial space algorithm which decides if a given annotated Θ-equation $t = t'$ with variables in $\mathbf{X}$ *fails* in some Conway theory. The result then follows by Savitch'a theorem [43]. By Theorem 3.2.3, it is enough to check if the annotated $\Sigma(X)$-equation $\mathrm{sig}_X^\#(t) = \mathrm{sig}_X^\#(t')$ fails in some Conway theory, or equivalently, if it fails in the free Conway theory $\mathcal{S}ch_{\Sigma(X)}/\equiv$. Let $\varphi : \mathcal{T}_\Theta(\Sigma(X)) \to \mathcal{S}ch_{\Sigma(X)}$ be the unique homomorphism mapping each symbol $\sigma \in \Sigma(X)_p$ to the corresponding atomic scheme $\widehat{\sigma} : 1 \to p$. Then $\mathrm{sig}_X^\#(t) = \mathrm{sig}_X^\#(t')$ fails in $\mathcal{S}ch_{\Sigma(X)}$ if and only if $\varphi(\mathrm{sig}_X^\#(t)) \not\equiv \varphi(\mathrm{sig}_X^\#(t'))$. The two schemes $\mathcal{S} := \varphi(\mathrm{sig}_X^\#(t))$ and $\mathcal{S}' := \varphi(\mathrm{sig}_X^\#(t'))$ can be constructed in polynomial space, as well as their minimal direct product $[_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$. By Theorem 4.1.27, our algorithm only has to check if $\mathcal{S}$ and $\mathcal{S}'$ are not strongly equivalent or if at least one of the two congruences $\ker_\pi$ or $\ker_{\pi'}$ is not aperiodic on $[_{\mathcal{S}}\Gamma_{\mathcal{S}'}]$. It can be decided in polynomial time if two schemes are not strongly equivalent, so the problem is reduced to testing if a congruence ρ is not aperiodic on a scheme $\mathcal{F}$. This can be done by guessing a congruence class $C \in \mathbf{Cl}(\rho)$, a nonsingleton subset $C' = \{c_1, \ldots, c_m\}$ of C, and a word $u \in [\omega]^*$ such that

$$u_\mathcal{F}(c_1) = c_2, \quad u_\mathcal{F}(c_2) = c_3, \quad \ldots \quad u_\mathcal{F}(c_{m-1}) = c_m, \quad u_\mathcal{F}(c_m) = c_1. \tag{4.4}$$

Let n be the number of states in $\mathcal{F}$. It is not allowed to store the whole word u, since it can be approximately as long as $\binom{n}{m} \cdot m!$. Instead, we guess u letter by letter and keep track only of its length, and the states c_i and $u_\mathcal{F}(c_i)$, $i \in [m]$. The procedure stops if condition (4.4) holds or $|u| > \binom{n}{m} \cdot m!$. □

Summary

This work contains results on two topics: the problem of deciding if a regular language is star-free, and the description of the free Conway theories.

The first chapter is a summary of the notations and some well known results which are used in the forthcoming chapters.

The second chapter contains the results of the paper [6] and a new result, Theorem 2.3.2, which was not published yet. Moreover, due to a minor modification of Construction 2.2.2, Theorem 2.3.1 is a slightly stronger than the corresponding theorem published in [6].

The results of the second chapter can be summarized as follows.

The intersection problem of nondeterministic automata (denoted **AIP**) asks to decide whether the intersection of the languages recognized by some given nondeterministic automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ $(n \geq 2)$ is not empty. It is proved in Theorem 2.3.1 that both this problem and the intersection problem of minimal 1-reset automata (denoted $\mathbf{AIP}_R$) is **PSPACE**-complete. This is achieved by showing that any problem in **PSPACE** is logspace-reducible to $\mathbf{AIP}_R$ (that is, $\mathbf{AIP}_R$ is **PSPACE**-hard), and that the more general problem **AIP** is solvable in polynomial space. The proof of the **PSPACE**-hardness of $\mathbf{AIP}_R$ is based on Construction 2.2.1, which is a sharpening of Kozen's construction [39] in the sense that Construction 2.2.1 yields a collection of much simpler finite automata, namely, 1-reset automata. It is proved in Theorem 2.3.2 that the intersection problem of complete reset automata is solvable in polynomial time.

The star-freeness problem of nondeterministic automata (denoted **ASF**) asks to decide whether the language recognized by a given nondeterministic automaton is star-free. It is proved in Theorem 2.3.3 that both this problem and the star-freeness problem of minimal deterministic automata having two input symbols (denoted $\mathbf{ASF}_R$) is **PSPACE**-complete. This is achieved by showing that the complement of $\mathbf{AIP}_R$ (that is, the problem of deciding whether the intersection of the languages recognized by some minimal 1-reset automata is empty) is logspace-reducible to $\mathbf{ASF}_R$, and that the more general problem **ASF** is solvable in polynomial space. It is not a new result that $\mathbf{ASF}_R$ is **PSPACE**-complete, since this was proved by Cho and Huynh in [20]. However, the reduction of the complement of $\mathbf{AIP}_R$ to $\mathbf{ASF}_R$ is simpler than the reduction given by Cho and Huynh due to the fact that the input of the problem $\mathbf{ASF}_R$ is a sequence of 1-reset automata, which are a very special

kind of aperiodic automata. (Cho and Huynh were using Kozen's aforementioned construction as a starting point in proving that the problem $\mathbf{ASF}_R$ is **PSPACE**-hard. Since the result of that construction is a sequence of automata which are not aperiodic in general, there is an additional step in the proof of Cho and Huynh in order to get a sequence of aperiodic automata. No such modification is needed in my argument.) It is a new result that the problem **ASF** is solvable in polynomial space.

The star-freeness problem of regular expressions (**RSF**) asks to decide whether the language denoted by a given regular expression is star-free. It is proved in Theorem 2.3.4 that both this problem and the star-freeness problem of regular expressions of star-height two over a two-element alphabet (denoted $\mathbf{RSF}_R$) is **PSPACE**-complete. This is achieved by showing that the complement of $\mathbf{AIP}_R$ is logspace-reducible to $\mathbf{RSF}_R$, and that the more general problem **RSF** is logspace-reducible to **ASF**. The former reduction is again heavily based on the fact that the problem $\mathbf{AIP}_R$ deals with a collection of 1-reset automata.

The third chapter is a summary of the notations and known results on many-sorted algebra.

The fourth chapter of the thesis contains the results of the paper [8].

Theorem 3.3.2 states that the nonstandard interpretations of flowchart schemes (that is, the theories enriched with an iteration operation satisfying all equations true of flowcharts) are exactly the Conway theories defined in [15]. Thus the least congruence on Σ-schemes whose quotient is a theory gives the free Conway theory.

Theorem 4.3.2 provides an explicit description of the free Conway theories. The description uses aperiodic simulations of flowchart schemes. It follows that the equations that hold in Conway theories are exactly the valid "group-free" equations of iteration theories.

The proof of Theorem 4.3.2 is based on Theorem 3.3.2, as well as two other auxiliary results, Corollary 4.1.30 and Lemma 4.3.1, which are interesting in themselves.

In order to prove these results we introduced and investigated three equivalence relations on the class of flowchart schemes, namely

1. The least congruence on the algebra of schemes such that the quotient is a preiteration theory.
2. The smallest equivalence identifying any two schemes $\mathcal{S}$ and $\mathcal{S}'$ such that there is a simple aperiodic homomorphisms from $\mathcal{S}$ to $\mathcal{S}'$.
3. The smallest equivalence identifying any two schemes $\mathcal{S}$ and $\mathcal{S}'$ such that there is an aperiodic homomorphism from $\mathcal{S}$ to $\mathcal{S}'$.

After proving a few technical lemmas—the most important being Lemma 4.1.29—it was established in Corollary 4.1.30 that the second of the above equivalences is the same as the third one.

A concrete description of the third equivalence is given in Theorem 4.1.27.

Lemma 4.3.1 states that the second of the above equivalences is the same as the first one. Consequently, a concrete description of the Σ-generated free Conway theory is the quotient of the algebra of Σ-schemes under the third equivalence relation, for any signature Σ.

Finally, the explicit description stated in Theorem 4.1.27 is used to prove in Theorem 4.3.3 that the equational theory of Conway theories is **PSPACE**-complete, as well as a few other problems on Σ-schemes, provided that the signature Σ contains at least one symbol of rank at least two.

Theorems 3.3.2 and 4.3.2 answer open problems raised in [13] and [15].

Összefoglalás

Jelen dogozat két témával kapcsolatos eredményeket tartalmaz: az egyik a reguláris nyelvek csillagmentességi problémája, a másik a szabad Conway elméletek leírása.

Az első fejezet a dolgozatban használt jelölések és néhány jól ismert eredmény összefoglalása.

A második fejezet a [6] cikkben publikált eredményeken kívül tartalmaz egy új, eddig nem publikált eredményt is, a 2.3.2. tételt, továbbá a 2.2.2. konstrukción végrehajtott kisebb módosításnak köszönhetően a 2.3.1. tétel állítása valamivel erősebb, mint a [6] cikkben közölt megfelelő tételé.

A második fejezet eredményei a következők.

A nemdeterminisztikus automaták metszet problémája (röviden **AIP**) azt kérdezi, hogy valamely adott $\mathcal{A}_1, \ldots, \mathcal{A}_n$ $(n \geq 2)$ nemdeterminisztikus automaták által felismert nyelvek metszete tartalmaz-e legalább egy elemet. A 2.3.1. tétel kimondja, hogy mind ez a probléma, mind pedig a minimális 1-reszet automaták metszet problémája (röviden $\mathbf{AIP}_R$) **PSPACE**-teljes. Ez abból következik, hogy minden **PSPACE**-beli probléma logaritmikus tárban visszavezethető az $\mathbf{AIP}_R$ problémára (vagyis $\mathbf{AIP}_R$ **PSPACE**-nehéz), és hogy az általánosabb **AIP** probléma polinomiális tárban megoldható. Az $\mathbf{AIP}_R$ probléma **PSPACE**-nehéz voltának igazolása a 2.2.1. konstrukción alapszik, amely Kozen [39] konstrukciójának javítása abban az értelemben, hogy a 2.2.1. konstrukció sokkal speciálisabb szerkezetű véges automatákat, nevezetesen 1-reszet automatákat szolgáltat eredményül. A 2.3.2. tétel kimondja, hogy a teljes reszet automaták metszet problémája polinom időben megoldható.

A nemdeterminisztikus automaták csillagmentességi problémája (röviden **ASF**) azt kérdezi, hogy valamely adott véges nemdeterminisztikus automata által felismert nyelv csillagmentes-e. A 2.3.3. tétel kimondja, hogy mind ez a probléma, mind pedig a két bemenőjeles minimális determinisztikus automaták csillagmentességi problémája (röviden $\mathbf{ASF}_R$) **PSPACE**-teljes. Ez abból következik, hogy az $\mathbf{AIP}_R$ probléma komplementere (vagyis az a probléma, amelyben el kell dönteni, hogy valamely adott minimális 1-reszet automaták által felismert nyelvek metszete üres-e) logaritmikus tárban visszavezethető az $\mathbf{ASF}_R$ problémára, és hogy az általánosabb **ASF** probléma polinomiális tárban megoldható. Az $\mathbf{ASF}_R$ probléma **PSPACE**-teljessége nem új eredmény, mivel azt Cho és Huynh [20] bizonyította először. Mindazonáltal az $\mathbf{AIP}_R$ probléma komplementerének az **ASF** problémára való általam adott visszavezetése egyszerűbb a Cho és Huynh által adott visszavezetésnél annak kö-

szönhetően, hogy az $\mathbf{ASF}_R$ probléma bemenő adatai minimális 1-reszet automaták, melyek igen speciális aperiodikus automaták. (Cho és Huynh Kozen előbb említett konstrukcióját használta kiindulópontként annak bizonyítására, hogy az $\mathbf{ASF}_R$ probléma **PSPACE**-teljes. Mivel ez a konstrukció általában nem aperiodikus automatákat ad eredményül, a bizonyításba be kellett iktatniuk egy módosító lépést, amely a konstrukció által szolgáltatott automatákat aperiodikussá alakítja. Az én bizonyításom nem tartalmaz ilyen lépést.) Az **ASF** probléma polinomiális tárban való megoldhatósága új eredmény.

A reguláris kifejezések csillagmentességi problémája (röviden **RSF**) azt kérdezi, hogy valamely adott reguláris kifejezés által jelölt nyelv csillagmentes-e. A 2.3.4. tétel kimondja, hogy mind ez a probléma, mind pedig a kételemű ábécé feletti kettő csillagmélységű reguláris kifejezések csillagmentességi problémája (röviden $\mathbf{RSF}_R$) **PSPACE**-teljes. Ez abból következik, hogy az $\mathbf{AIP}_R$ probléma komplementere logaritmikus tárban visszavezethető az $\mathbf{RSF}_R$ problémára, és hogy az általánosabb **RSF** probléma logaritmikus tárban visszavezethető az **ASF** problémára. Az előbbi visszavezetés ismétcsak erősen támaszkodik arra a tényre, hogy az $\mathbf{AIP}_R$ probléma bemenő adatai 1-reszet automaták.

A harmadik fejezet a többtípusú algebrákkal kapcsolatos jelölések és néhány ismert eredmény összefoglalása.

A dolgozat negyedik fejezete a [8] cikk eredményeit tartalmazza.

A 3.3.2. tétel kimondja, hogy a folyamatábra sémák nem-standard interpretációi (vagyis azok az iteráció művelettel kibővített algebrai elméletek, amelyek kielégítik a folyamatábra sémákra teljesülő azonosságok mindegyikét) éppen a [15] könyvben definiált Conway elméletek. Következésképpen a Σ szignatúra által generált szabad Conway elmélet a Σ-sémák legszűkebb olyan kongruencia melletti hányadosa, amely melletti hányados algebrai elmélet.

A 4.3.2. tétel megadja a szabad Conway elméletek egy explicit leírását. A leírás a folyamatábra sémák aperiodikus szimulációin alapul. A tételből következik, hogy a Conway elméletekben teljesülő azonosságok éppen az iterációs elméletek "csoportmentes" azonosságai.

A 4.3.2. tétel a 3.3.2. tételből és két további állításból, a 4.1.30. következményből és a 4.3.1. segédtételből következik, melyek önmagukban is érdekes eredmények.

Ezen állítások igazolásához bevezettük és megvizsgáltuk az alábbi három ekvivalencia relációt a folyamatábra sémákon.

1. A folyamatábra sémák algebrájának legszűkebb olyan kongruenciája, amely melletti hányados preiterációs elmélet.

2. A legszűkebb olyan ekvivalencia, amely azonosít bármely két $\mathcal{S}$ és $\mathcal{S}'$ sémát, melyekre létezik egyszerű aperiodikus homomorfizmus $\mathcal{S}$-ből $\mathcal{S}'$-be.

3. A legszűkebb olyan ekvivalencia, amely azonosít bármely két $\mathcal{S}$ és $\mathcal{S}'$ sémát, melyekre létezik aperiodikus homomorfizmus $\mathcal{S}$-ből $\mathcal{S}'$-be.

Néhány technikai jellegű segédtétel bizonyítása után – melyek közül a 4.1.29. segédtétel a legfontosabb – a 4.1.30. következmény kimondja, hogy a fenti ekvivalenciák közül a második megegyezik a harmadikkal.

A 4.1.27. tétel megadja a harmadik ekvivalencia egy konkrét leírását.

A 4.3.1. segédtétel kimondja, hogy a fenti ekvivalenciák közül a második megegyezik az elsővel. Következésképpen tetszőleges Σ szignatúrára a Σ által generált szabad Conway elmélet leírható úgy, mint a Σ-sémák algebrájának a harmadik ekvivalencia melletti hányadosa.

Végezetül a 4.1.27. tételben adott explicit leírásra támaszkodva a 4.3.3. tétel kimondja, hogy a Conway elméletek azonosságelmélete **PSPACE**-teljes, csakúgy, mint néhány további, a Σ-sémákra vonatkozó probléma abban az esetben, ha Σ tartalmaz olyan szimbólumot, melynek aritása legalább kettő.

A 3.3.2. és 4.3.2. tételek a [13] cikkben, illetve a [15] könyvben felvetett kérdésekre adnak választ.

Bibliography

[1] M. Bartha. An equational axiomatization of systolic systems. *Theoretical Computer Science*, 55:265–289, 1987.

[2] M. Bartha. Interpretations of synchronous flowchart schemes. In J. Csirik and F. Gécseg, editors, *7th Conference on the Fundamentals of Computation Theory, Szeged*, volume 380 of *Lecture Notes in Computer Science*, pages 25–34, Berlin, 1989. Springer-Verlag.

[3] M. Bartha. An algebraic model of synchronous systems. *Information and Computation*, 97:97–131, 1992.

[4] M. Bartha. Foundations of a theory of synchronous systems. *Theoretical Computer Science*, 100:325–346, 1992.

[5] J. A. Bergstra and G. Stefanescu. Processes with multiple entries and exits modulo isomorphism and bisimulation. *Fundamenta Informaticae*, 27:37–56, 1996.

[6] L. Bernátsky. Regular expression star-freeness is PSPACE-complete. *Acta Cybernetica*, 13:1–21, 1997.

[7] L. Bernátsky, S. L. Bloom, Z. Ésik, and G. Stefanescu. Equational theories of relations and regular sets, Extended abstract. In *Proceedings of the 2nd Colloquium on Words, Combinatorics and Semigroups*, 1992.

[8] L. Bernátsky and Z. Ésik. Semantics of flowchart programs and the free Conway theories. *Theoretical Informatics and Applications*, 32(1–2–3):35–78, 1998.

[9] G. Birkhoff. On the structure of abstract algebras. *Proc. Camb. Philos. Soc.*, 31:433–454, 1935.

[10] G. Birkhoff and J. D. Lipson. Heterogeneous algebras. *Journal of Combinatorial Theory*, 8:115–133, 1970.

[11] S. L. Bloom, C. C. Elgot, and J. B. Wright. Solutions of the iteration equation and extensions of the scalar iteration operation. *SIAM Journal of Computing*, 9:24–65, 1980.

[12] S. L. Bloom, C. C. Elgot, and J. B. Wright. Vector iteration in pointed iterative theories. *SIAM Journal of Computing*, 9:525–540, 1980.

[13] S. L. Bloom and Z. Ésik. Axiomatizing schemes and their behaviours. *Journal of Computing and System Sciences*, 31:375–393, 1985.

[14] S. L. Bloom and Z. Ésik. Floyd–Hoare logic in iteration theories. *JACM*, 38:887–934, 1991.

[15] S. L. Bloom and Z. Ésik. *Iteration theories: the equational logic of iterative processes.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1993.

[16] S. L. Bloom and Z. Ésik. The equational logic of fixed points. *Theoretical Computer Science*, 179(1–2):1–60, 1997.

[17] D. P. Bovet and P. Crescenzi. *Introduction to the theory of complexity.* Prentice Hall, 1994.

[18] S. Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics.* Springer, New York, 1981.

[19] V. E. Cazanescu and G. Stefanescu. Towards a new algebraic foundation of flowchart scheme theory. *Fundamenta Informaticae*, 13:171–210, 1990.

[20] S. Cho and D. T. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88:99–116, 1991.

[21] J. C. Conway. *Regular algebra and finite machines.* Chapman and Hall, 1971.

[22] B. Courcelle. Fundamental properties of infinite trees. In *Theoretical foundations of programming methodology, Munich 1981.* Reidel, 1982.

[23] H. Ehrig and B. Mahr. Theory and practice of software development: A review of driving forces and expectations. *Bulletin of the European Association for Theoretical Computer Science*, 57, 1995.

[24] S. Eilenberg. *Automata, languages and machines.* Academic Press, New York and London, 1974.

[25] C. C. Elgot. Monadic computation and iterative algebraic theories. In J. Shepherdson, editor, *Logic Colloquium 1973*, volume 80 of *Studies in Logic*, Amsterdam, 1975. North Holland.

[26] C. C. Elgot. Matricial theories. *Journal of Algebra*, 42:391–421, 1976.

[27] C. C. Elgot. Structured programming with and without goto statements. *IEEE Transactions on Software Engineering*, 2(1):41–54, Mar. 1976.

[28] C. C. Elgot, S. L. Bloom, and R. Tindell. Algebraic and graph theoretic characterizations of structured flowchart schemes. *Theoretical Computer Science*, 9:265–286, 1979.

[29] Z. Ésik. Identities in iterative and rational algebraic theories. *Computational Linguistics and Computer Languages*, 14:183–207, 1980.

[30] Z. Ésik. Group axioms for iteration. *Information and Computation*, 148:131–180, 1999.

[31] Z. Ésik and L. Bernátsky. Equational properties of Kleene algebras of relations with conversion. *Theoretical Computer Science*, 137:237–251, 1995.

[32] Z. Ésik and L. Bernátsky. Scott induction and equational proofs. In *MFPS XI, 1995*, volume 1 of *Electronic Notes in Theoretical Computer Science (ENTCS)*, 1995. http://www.elsevier.nl/locate/entcs/volume1/esik.ps, 28 pages.

[33] M. R. Garey and D. S. Johnson. *Computers and intractability, A guide to the theory of NP-completeness.* W. H. Freeman and Company, New York, 1979.

[34] J. Goguen and J. Meseguer. Completeness of many-sorted equational logic. *Houston Journal of Mathematics*, 11:307–334, 1985.

[35] J. S. Golan. *The theory of semirings with applications in mathematics and theoretical computer science.* Longman Scientific & Technical, 1993.

[36] I. Guessarian. Many-sorted logics and algebraic semantics. In K. Meinke and J. V. Tucker, editors, *Many-sorted logic and its applications*, pages 123–134. John Wiley & Sons Ltd., 1993.

[37] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers.* Oxford University Press, London, 3rd edition, 1954.

[38] P. J. Higgins. Algebras with a scheme of operators. *Mathematische Nachrichten*, 27:115–132, 1963.

[39] D. Kozen. Lower bounds for natural proof systems. In *Proc. 18th Ann. Symp. on Foundations of Computer Science*, pages 254–266, Long Beach, CA, 1977. IEEE Computer Society.

[40] D. Krob. Complete systems of B-rational identities. *Theoretical Computer Science*, 89:207–343, 1991.

[41] F. W. Lawvere. Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences USA*, 50:869–873, 1963.

[42] V. Manca and A. Salibra. Soundness and completeness of the Birkhoff equational calculus for many-sorted algebras with possibly empty carrier sets. *Theoretical Computer Science*, 94(1):101–124, Mar. 1992.

[43] C. H. Papadimitriou. *Computational complexity.* Addison-Wesley, New York, 1994.

[44] J.-E. Pin. *Varieties of formal languages.* North Oxford Academic, 1986.

[45] A. Salomaa. *Theory of automata.* Pergamon Press, 1969.

[46] W. J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computing and System Sciences*, 4, 1970.

[47] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[48] G. Stefanescu. On flowchart theories: part I. The deterministic case. *Journal of Computing and System Sciences*, 35:163–191, 1987.

[49] J. Stern. Complexity of some problems from the theory of automata. *Information and Control*, 66:163–176, 1985.

[50] H. Straubing. *Finite automata, formal languages and circuit complexity.* Birkhäuser, 1994.

[51] M. Wirsing. Algebraic specification: semantics, parametrization and refinement. In E. J. Neuhold and M. Paul, editors, *Formal Description of Programming Concepts*, IFIP State-of-the-Art Reports, pages 259–318. Springer-Verlag, 1991.

Index

List of Symbols

List of Figures